Autonomous and Mobile Robotics

Prof. Giuseppe Oriolo

Wheeled Mobile Robots

Motion Control: Introduction and Trajectory Tracking

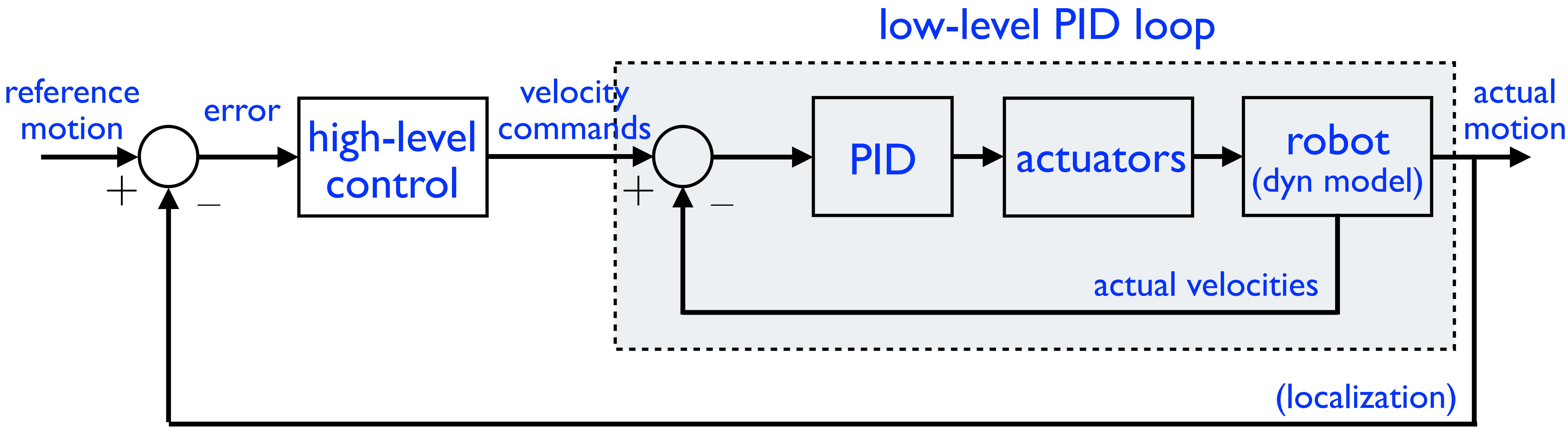
Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

Sapienza Università di Roma

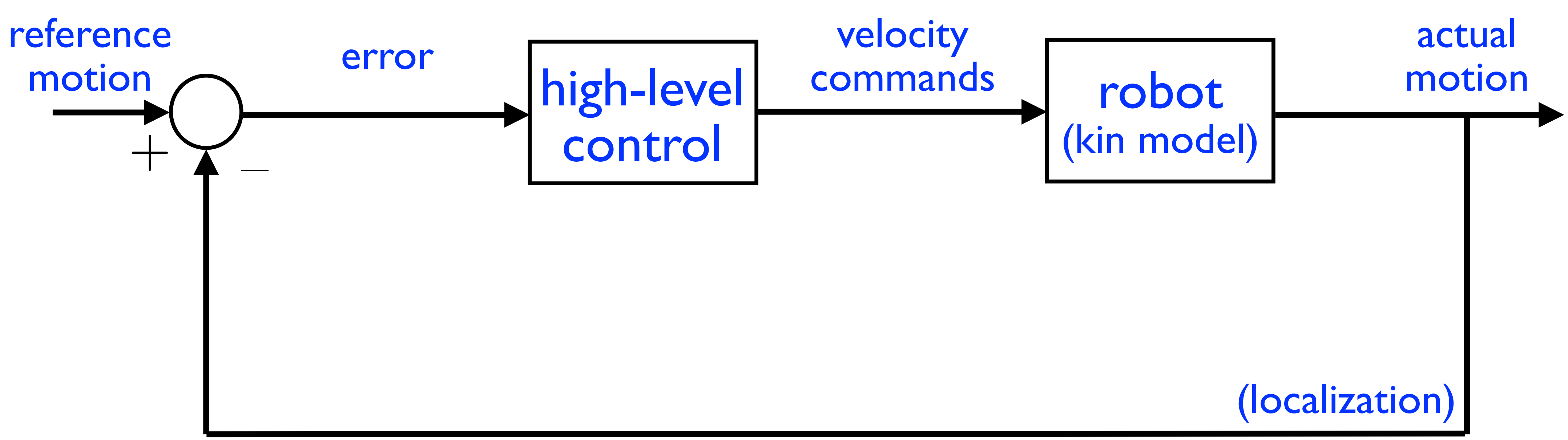
motion control

- a desired motion is assigned for the WMR, and the associated nominal inputs have been computed
- to execute the desired motion, we need feedback control because the application of nominal inputs in open-loop would lead to very poor performance
- in manipulators, we use dynamic models to compute commands at the generalized force level
- in WMRs, we use kinematic models because (1) wheels are equipped with low-level PID loops that accept velocities as reference (2) dynamics is simpler and can be mostly canceled via feedback

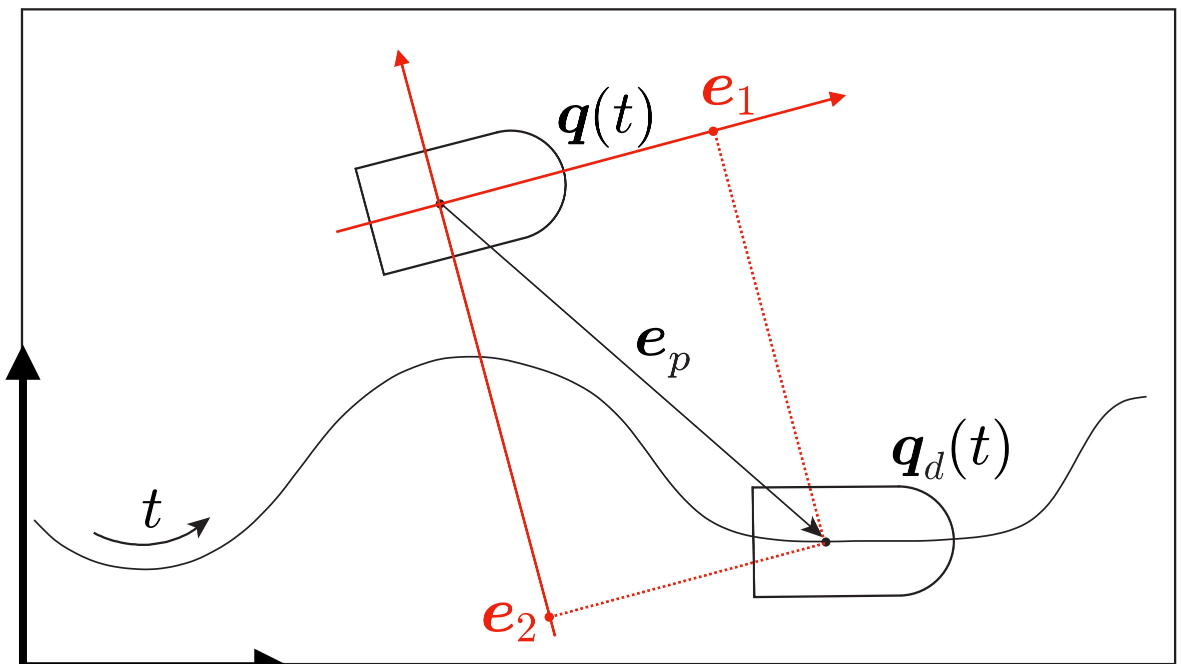
- actual control scheme



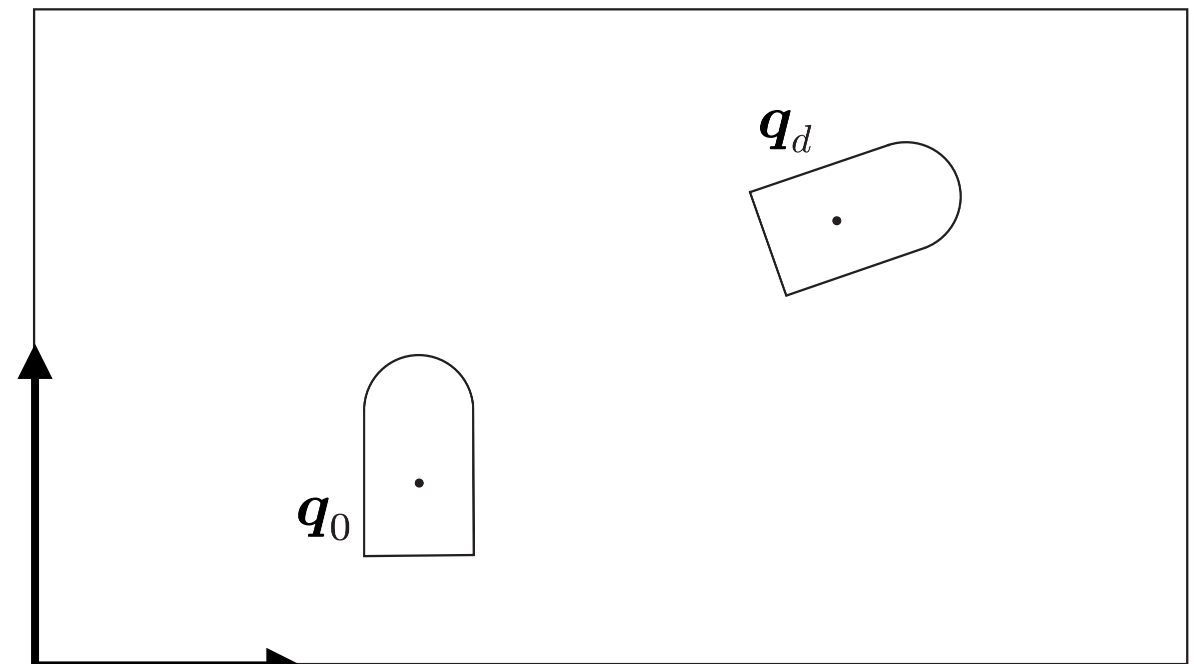
- equivalent control scheme (for design)



motion control problems



trajectory tracking
(predictable transients)



posture regulation
(no prior planning)

- other problems of interest
  - path tracking (only geometric motion)
  - Cartesian regulation (final orientation is free)
- w.l.o.g., we consider a unicycle in the following

trajectory tracking: state error feedback

- the unicycle must track a Cartesian desired trajectory $(x_d(t), y_d(t))$ that is admissible, i.e., there exist v_d and ω_d such that

$$\dot{x}_d = v_d \cos\theta_d$$
$$\dot{y}_d = v_d \sin\theta_d$$
$$\dot{\theta}_d = \omega_d$$

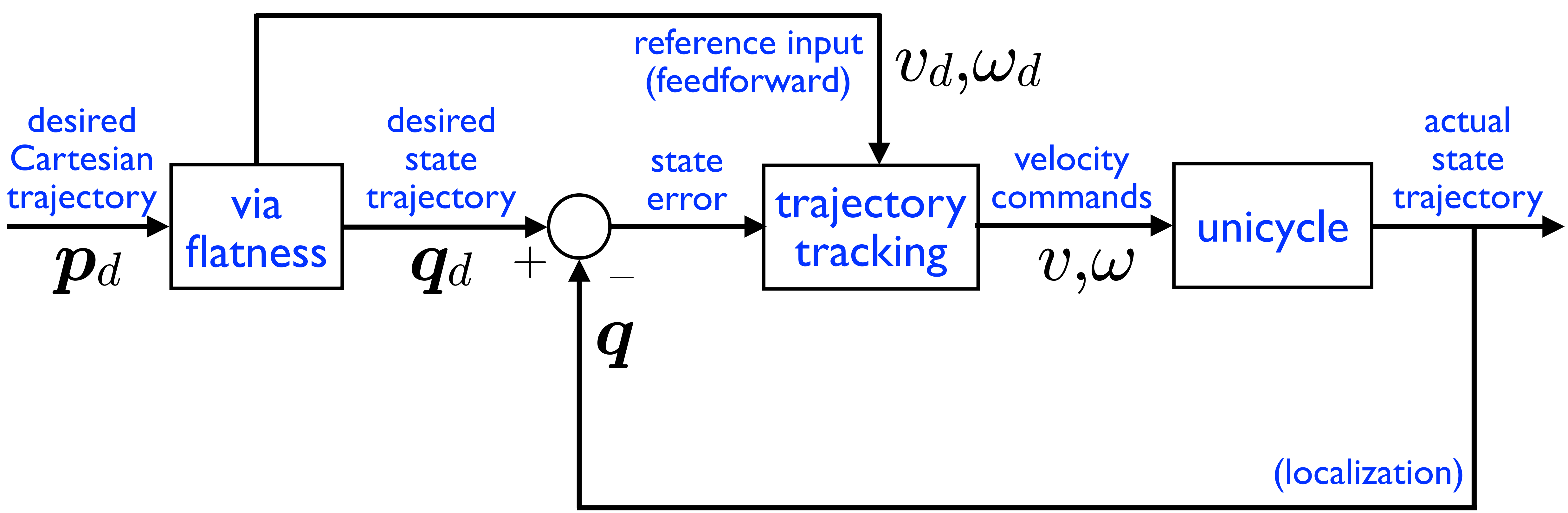
- thanks to flatness, from $(x_d(t), y_d(t))$ we can compute

$$\theta_d(t) = \mathrm{Atan2}\left(\dot{y}_d(t), \dot{x}_d(t)\right) + k\pi \qquad k = 0, 1$$

$$v_d(t) = \pm\sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)}$$

$$\omega_d(t) = \frac{\ddot{y}_d(t)\dot{x}_d(t) - \ddot{x}_d(t)\dot{y}_d(t)}{\dot{x}_d^2(t) + \dot{y}_d^2(t)}$$

- admissibility of the trajectory is guaranteed as long as $\dot{y}_d/\dot{x}_d$ is continuous in t (no sharp corners)
- the desired state trajectory can be used to compute the state error, from which the feedback action is generated; whereas the nominal input can be used as a feedforward term
- the resulting block scheme will be



- rather than using directly the state error $\boldsymbol{q}_d - \boldsymbol{q}$, use its rotated version defined as

$$\boldsymbol{e} = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{pmatrix}$$

  (e_1, e_2) is the Cartesian error $\boldsymbol{e}_p$ in a frame rotated by θ (in red in slide 4)

- the error dynamics is nonlinear and time-varying

$$\dot{e}_1 = v_d \cos e_3 - v + e_2\,\omega$$
$$\dot{e}_2 = v_d \sin e_3 - e_1\,\omega$$
$$\dot{e}_3 = \omega_d - \omega$$

approximate linearization: brush-up

- idea: to stabilize a nonlinear system at an equilibrium, stabilize its approximate linearization around it
- for a generic nonlinear system

$$\dot{\boldsymbol{x}} = \boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \qquad \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{u} \in \mathbb{R}^m$$

assume the origin (w.l.o.g.) is the desired unforced equilibrium, so that

$$\boldsymbol{\varphi}(\mathbf{0}, \mathbf{0}) = \mathbf{0} \qquad \text{or equivalently} \qquad \boldsymbol{f}(\mathbf{0}) = \mathbf{0}$$

- the Taylor expansion of $\boldsymbol{\varphi}$ around $\boldsymbol{x} = \mathbf{0}, \boldsymbol{u} = \mathbf{0}$ gives

$$\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{u}) \approx \cancel{\boldsymbol{\varphi}(\mathbf{0}, \mathbf{0})} + \left.\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x} = \mathbf{0} \\ \boldsymbol{u} = \mathbf{0}}} \boldsymbol{x} + \left.\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x} = \mathbf{0} \\ \boldsymbol{u} = \mathbf{0}}} \boldsymbol{u}$$

- the approximate linearization of the nonlinear system at the origin is then defined as

$$\dot{\boldsymbol{x}} = \left.\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{0}\\ \boldsymbol{u}=\boldsymbol{0}}} \boldsymbol{x} + \left.\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{0}\\ \boldsymbol{u}=\boldsymbol{0}}} \boldsymbol{u} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

- now let $\boldsymbol{u} = \boldsymbol{K}\boldsymbol{x}$, so that

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{K}\boldsymbol{x} = (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})\boldsymbol{x}$$

  if $\boldsymbol{K}$ is chosen in such a way that $\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}$ is Hurwitz (certainly possible if $(\boldsymbol{A},\boldsymbol{B})$ is controllable), then the approximate linearization is asymptotically stable

- by Lyapunov indirect method, this ensures the origin is locally asymptotically stable for the nonlinear system

via approximate linearization

- apply the approximate linearization approach to stabilize the previous error dynamics

$$\begin{aligned}\dot{e}_1 &= v_d \cos e_3 - v + e_2\,\omega \\ \dot{e}_2 &= v_d \sin e_3 - e_1\,\omega \\ \dot{e}_3 &= \omega_d - \omega\end{aligned}$$

- in this form, the origin is not an unforced equilibrium; however, this is easily rectified by using the following (invertible) input transformation

$$\begin{aligned}u_1 &= v_d \cos e_3 - v \\ u_2 &= \omega_d - \omega\end{aligned}$$

- we obtain

$$\dot{e}_1 = \omega_d\, e_2 + u_1 - e_2\, u_2$$
$$\dot{e}_2 = -\omega_d\, e_1 + v_d \sin e_3 + e_1\, u_2$$
$$\dot{e}_3 = u_2$$

that is, $\dot{\boldsymbol{e}} = \boldsymbol{\varphi}(\boldsymbol{e}, \boldsymbol{u})$ with $\boldsymbol{\varphi}(\mathbf{0}, \mathbf{0}) = \mathbf{0}$

- note that

$$\boldsymbol{\varphi}(\boldsymbol{e}, \boldsymbol{u}) = \overbrace{\begin{pmatrix} \omega_d\, e_2 \\ -\omega_d\, e_1 + v_d \sin e_3 \\ 0 \end{pmatrix}}^{\boldsymbol{f}(\boldsymbol{e})} + \overbrace{\begin{pmatrix} 1 & -e_2 \\ 0 & e_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}}^{\boldsymbol{G}(\boldsymbol{e})\boldsymbol{u}}$$

- the linear approximation of the error dynamics is

$$\dot{\boldsymbol{e}} = \left. \frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{e}} \right|_{\substack{\boldsymbol{e} = \mathbf{0} \\ \boldsymbol{u} = \mathbf{0}}} \boldsymbol{e} + \left. \frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{u}} \right|_{\substack{\boldsymbol{e} = \mathbf{0} \\ \boldsymbol{u} = \mathbf{0}}} \boldsymbol{u} = \boldsymbol{A}\boldsymbol{e} + \boldsymbol{B}\boldsymbol{u}$$

- one easily finds

$$\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{e}} = \begin{pmatrix} 0 & \omega_d - u_2 & 0 \\ -\omega_d + u_2 & 0 & v_d \cos e_3 \\ 0 & 0 & 0 \end{pmatrix} \qquad \frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{u}} = \begin{pmatrix} 1 & -e_2 \\ 0 & e_1 \\ 0 & 1 \end{pmatrix}$$

- letting $\boldsymbol{e} = \mathbf{0}, \boldsymbol{u} = \mathbf{0}$ gives

$$\boldsymbol{A} = \begin{pmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & v_d \\ 0 & 0 & 0 \end{pmatrix} \qquad \boldsymbol{B} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$

note that $\boldsymbol{A}$ is actually $\boldsymbol{A}(t)$ due to the dependence of the references inputs v_d and ω_d on time $\Rightarrow$ the linear approximation will be a time-varying system!

- wrapping up, the linearized approximation of the error dynamics around the reference trajectory is

$$\dot{\boldsymbol{e}} = \begin{pmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & v_d \\ 0 & 0 & 0 \end{pmatrix} \boldsymbol{e} + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

- now define the linear feedback

$$\boldsymbol{u} = \boldsymbol{K}\boldsymbol{e} = \begin{pmatrix} -k_1 & 0 & 0 \\ 0 & -k_2 & -k_3 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix}$$

- the closed-loop error dynamics is still time-varying!

$$\dot{\boldsymbol{e}} = (\boldsymbol{A}(t) + \boldsymbol{B}\boldsymbol{K})\boldsymbol{e} = \boldsymbol{A}_{cl}(t)\boldsymbol{e} = \begin{pmatrix} -k_1 & \omega_d & 0 \\ -\omega_d & 0 & v_d \\ 0 & -k_2 & -k_3 \end{pmatrix} \boldsymbol{e}$$

- letting

$$k_1 = k_3 = 2\zeta a \qquad k_2 = \frac{a^2 - \omega_d^2}{v_d}$$

with $a>0$, $\zeta\in(0,1)$, the characteristic polynomial of $\boldsymbol{A}(t)$ becomes time-invariant and Hurwitz

$$p(\lambda) = \underbrace{(\lambda + 2\zeta a)}_{\text{real negative eigenvalue}}\underbrace{(\lambda^2 + 2\zeta a\lambda + a^2)}_{\text{complex eigenvalues with damping } \zeta \text{ and natural frequency } a}$$

- caveat: this does not guarantee asymptotic stability, unless v_d and ω_d are constant (rectilinear and circular trajectories); even in this case, asymptotic stability of the unicycle is not global (indirect Lyapunov method)

- the actual velocity inputs v,ω are obtained plugging the feedback controls u_1, u_2 in the input transformation

- note: $(v,\omega) \to (v_d,\omega_d)$ as $\boldsymbol{e} \to \mathbf{0}$ (pure feedforward)

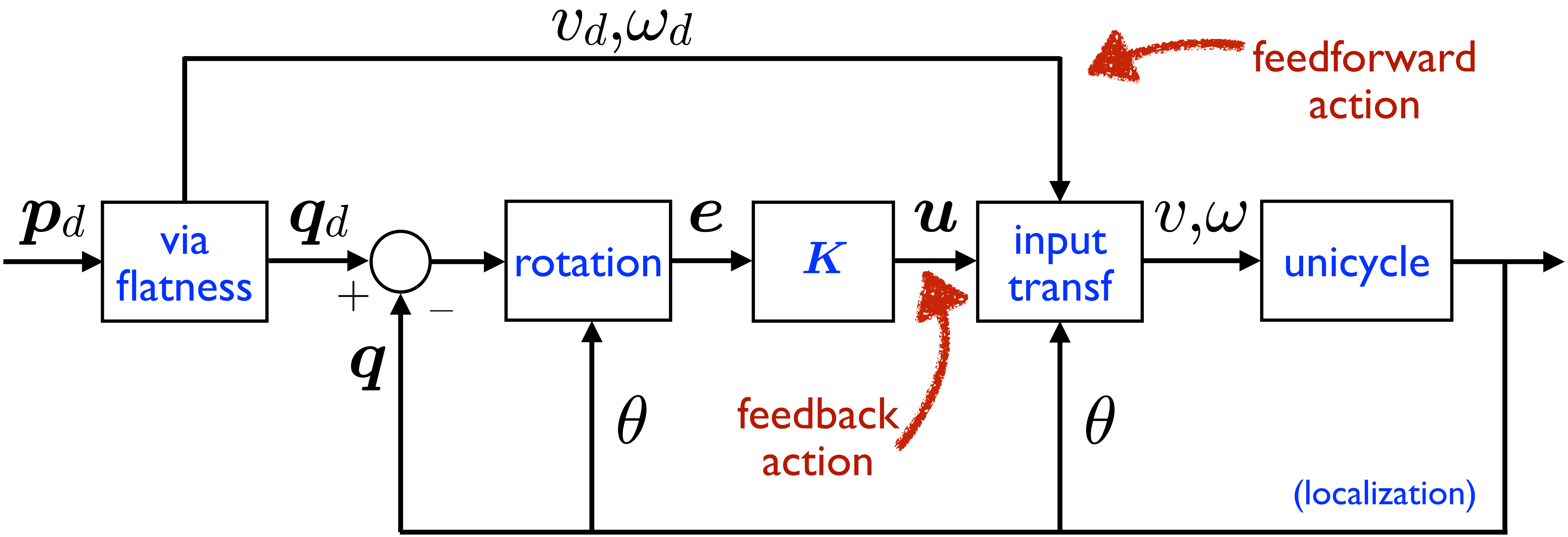
- note: $k_2 \to \infty$ as $v_d \to 0$, hence this controller can only be used with persistent Cartesian trajectories (stops are not allowed)

- global stability is guaranteed by a nonlinear version

$$
\begin{aligned}
u_1 &= -k_1(v_d,\omega_d)\, e_1 \\
u_2 &= -k_2\, v_d\, \frac{\sin e_3}{e_3}\, e_2 - k_3(v_d,\omega_d)\, e_3
\end{aligned}
$$

if k_1,k_3 bounded, positive, with bounded derivatives

- the final block scheme for trajectory tracking via state error feedback and approximate linearization is

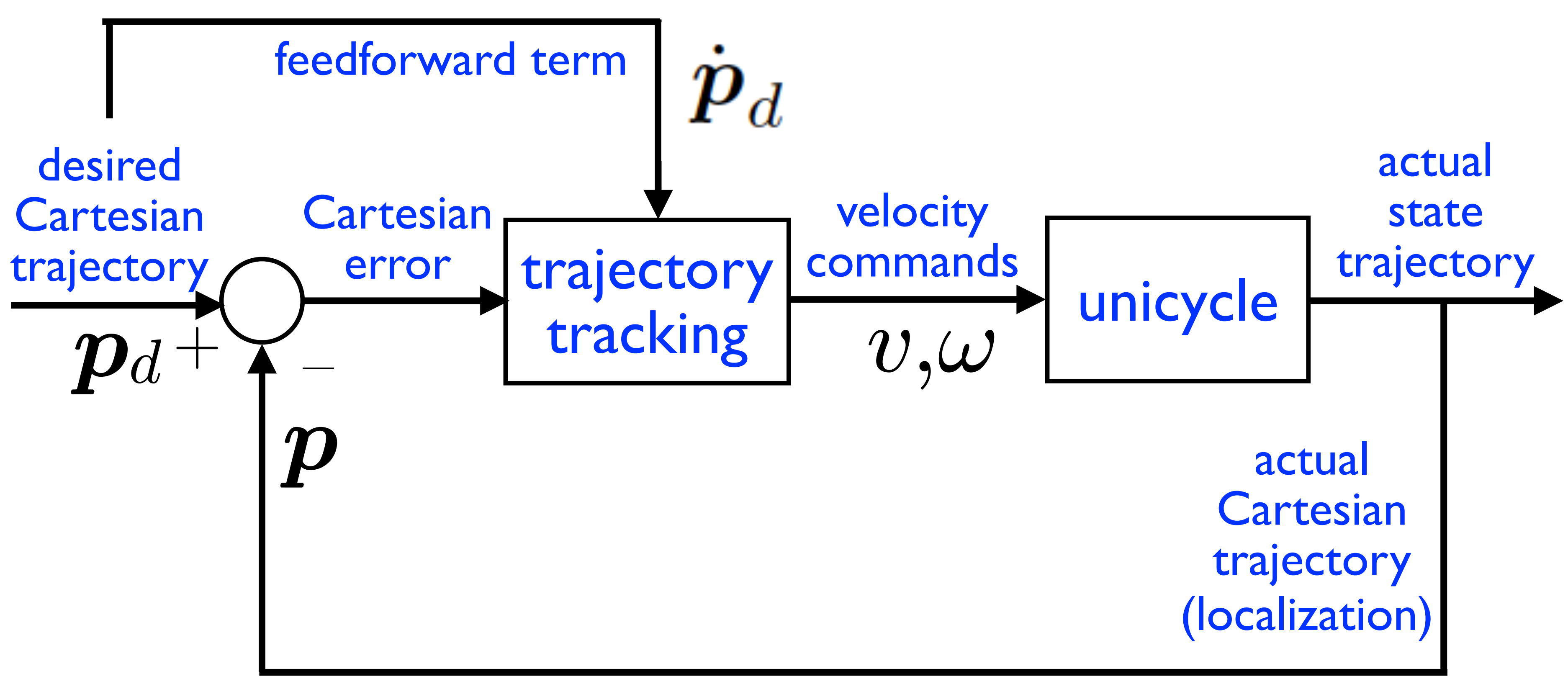


- a static controller based on state error
- needs v_d, ω_d
- needs θ also for error rotation + input transformation

trajectory tracking: output error feedback

- another approach: develop the feedback action from the output (Cartesian) error only, without computing a desired state trajectory, while the feedforward term is the velocity along the reference trajectory

- the resulting block scheme will be



exact i/o linearization: brush-up

- consider a driftless nonlinear system

$$\dot{\boldsymbol{x}} = \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \qquad \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{u} \in \mathbb{R}^m,\ \boldsymbol{y} \in \mathbb{R}^m$$
$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x})$$

- since

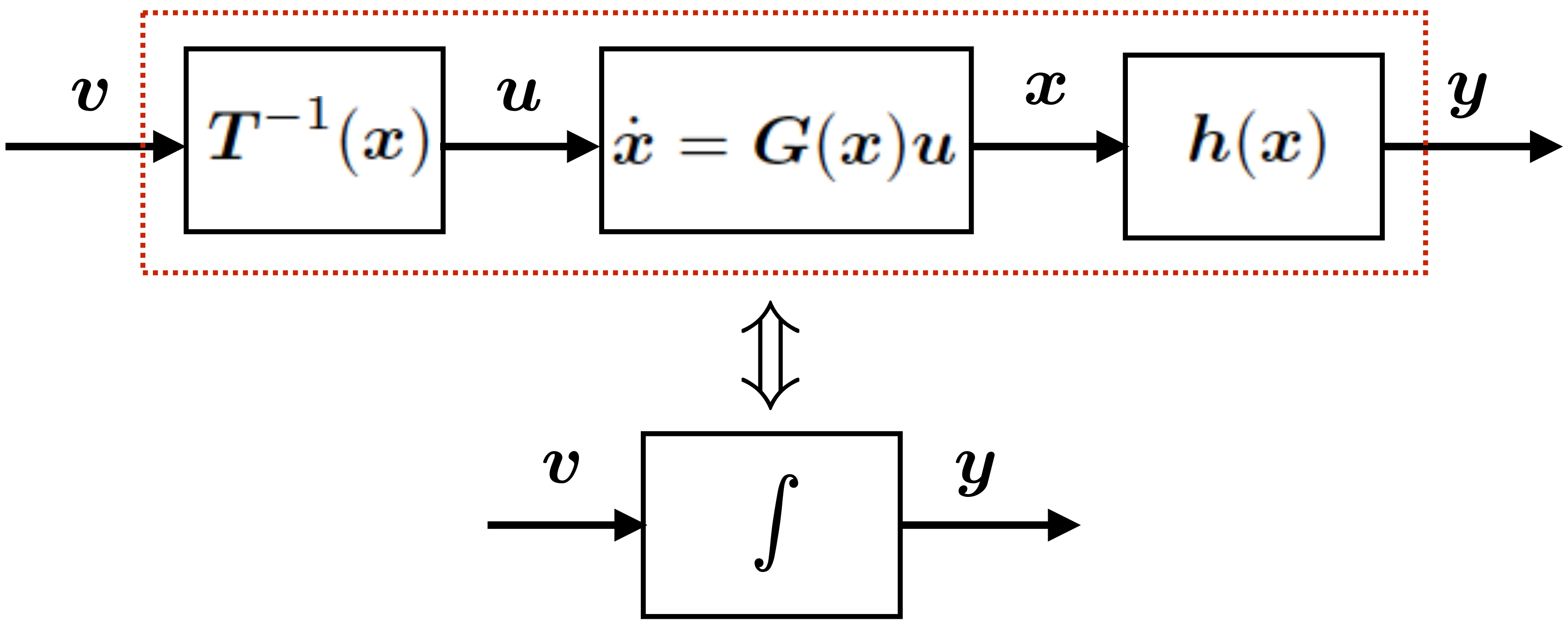
$$\dot{\boldsymbol{y}} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{T}(\boldsymbol{x})\boldsymbol{u}$$

if $\boldsymbol{T}$ ($m \times m$ decoupling matrix) is invertible we can set

$\boldsymbol{u} = \boldsymbol{T}^{-1}(\boldsymbol{x})\boldsymbol{v}$ obtaining $\dot{\boldsymbol{y}} = \boldsymbol{T}(\boldsymbol{x})\boldsymbol{T}^{-1}(\boldsymbol{x})\boldsymbol{v} = \boldsymbol{v}$

i.e., an exactly linear map between the new inputs $\boldsymbol{v}$ and (the time derivative of) the outputs

- in pictures



under the action
of the linearizing feedback
$u = T^{-1}(x)v\ldots$

the system behaves as

…a simple (vector) integrator
from v to y

- given a reference output $y_d(t)$, the dynamics of the output error $e = y_d - y$ is $\dot{e} = \dot{y}_d - \dot{y} = \dot{y}_d - v$
- let $v = \dot{y}_d + Ke$ (feedforward+proportional feedback) to obtain $\dot{e} = -Ke$, i.e., global exponential stability provided that the eigenvalues of K are in the rhp
- the final control law is $u = T^{-1}(x)(\dot{y}_d + Ke)$

via i/o linearization (static feedback)

- let us adopt the exact i/o linearization approach to design a Cartesian trajectory tracking controller for the unicycle

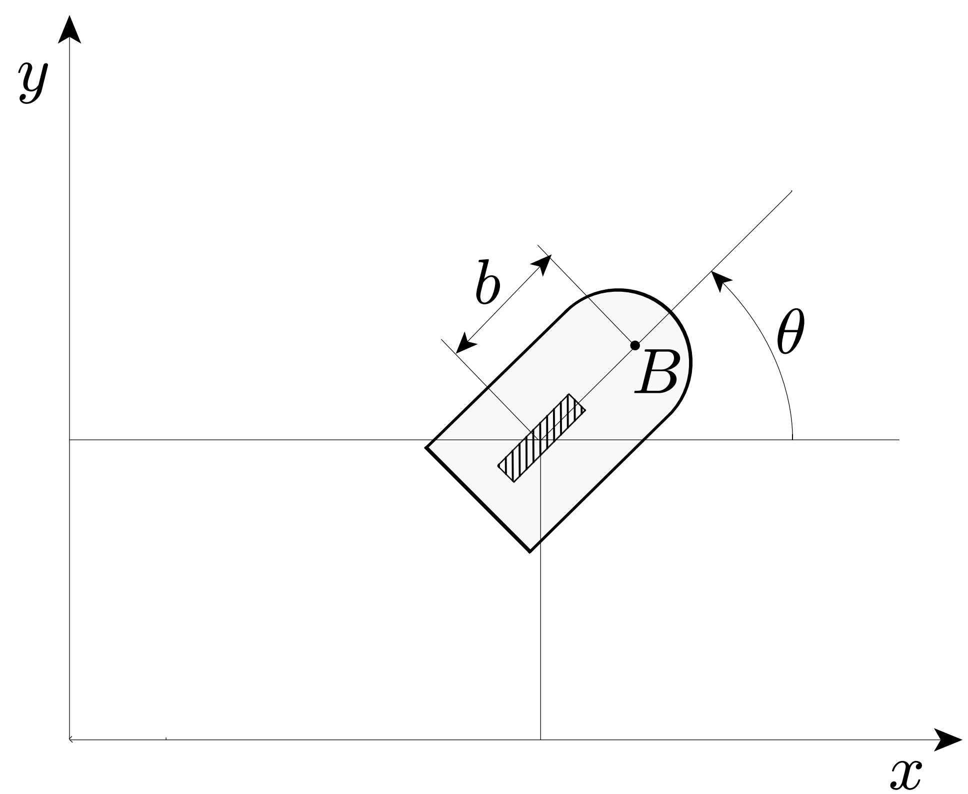
- however, in this case the decoupling matrix associated to the Cartesian position turns out to be singular

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}$$

as a consequence, exact input-output linearization is not possible for the output (x, y)

- solution: change slightly the output so that the new input-output map is invertible and exact linearization becomes possible

- displace the output from the contact point of the wheel to point B along the sagittal axis



$$y_1 = x_B = x + b\cos\theta$$
$$y_2 = y_B = y + b\sin\theta$$

- differentiating wrt time

$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} \cos\theta & -b\sin\theta \\ \sin\theta & b\cos\theta \end{pmatrix}}_{\det = b} \begin{pmatrix} v \\ \omega \end{pmatrix} = \boldsymbol{T}(\theta) \begin{pmatrix} v \\ \omega \end{pmatrix}$$

- if $b \neq 0$, we may set

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \boldsymbol{T}^{-1}(\theta) \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta/b & \cos\theta/b \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

obtaining an input-output linearized system

$$\begin{aligned} \dot{y}_1 &= u_1 \\ \dot{y}_2 &= u_2 \\ \dot{\theta} &= \frac{u_2\cos\theta - u_1\sin\theta}{b} \end{aligned}$$

- we achieve global exponential convergence of x_B, y_B to the desired trajectory by letting

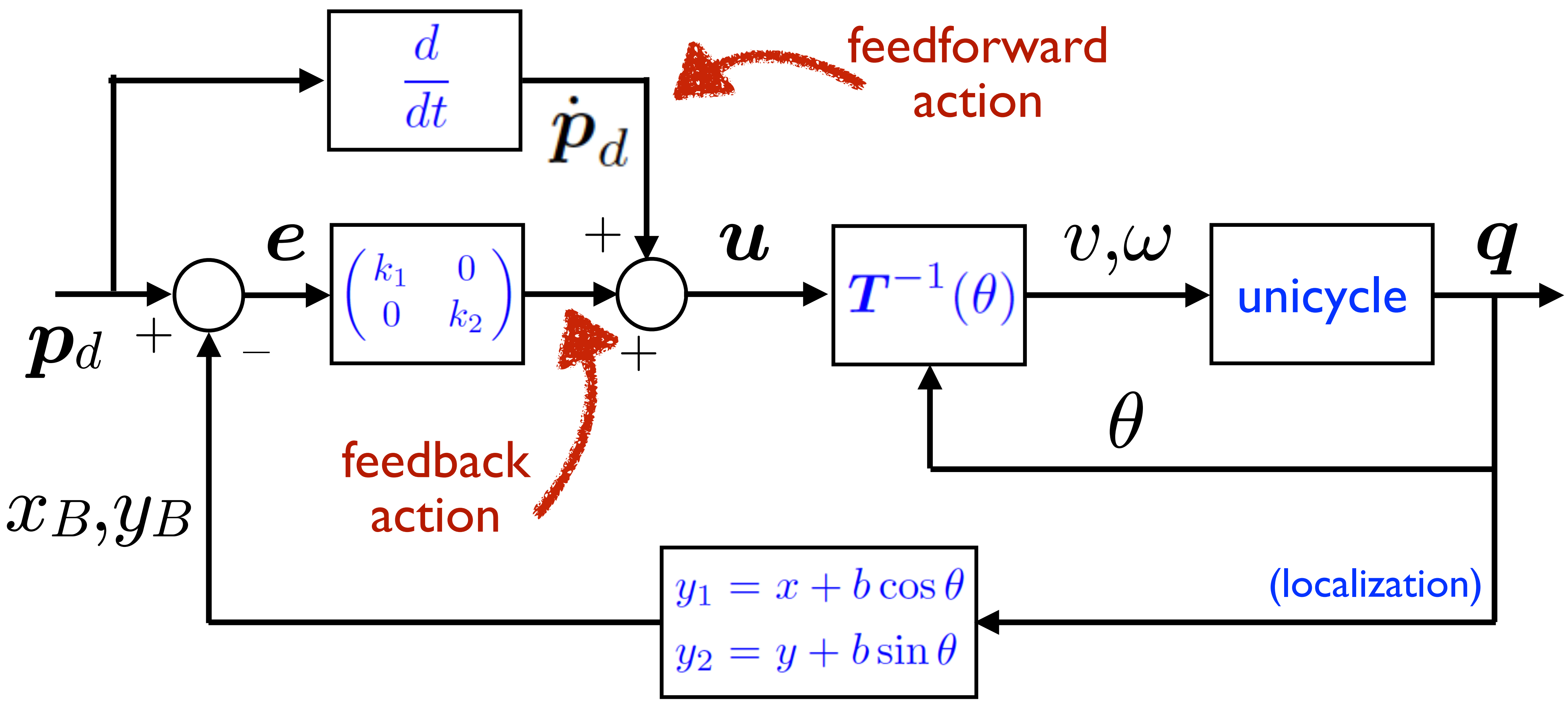
$$u_1 = \dot{x}_d + k_1(x_d - x_B)$$
$$u_2 = \dot{y}_d + k_2(y_d - y_B)$$

with $k_1, k_2 > 0$

- θ is not controlled with this scheme, which is based on output error feedback (there is a zero dynamics); still, it must evolve as dictated by flatness…

- the desired trajectory for B can be arbitrary; in particular, square corners may be included

- the final block scheme for trajectory tracking via output error feedback + static i/o linearization is

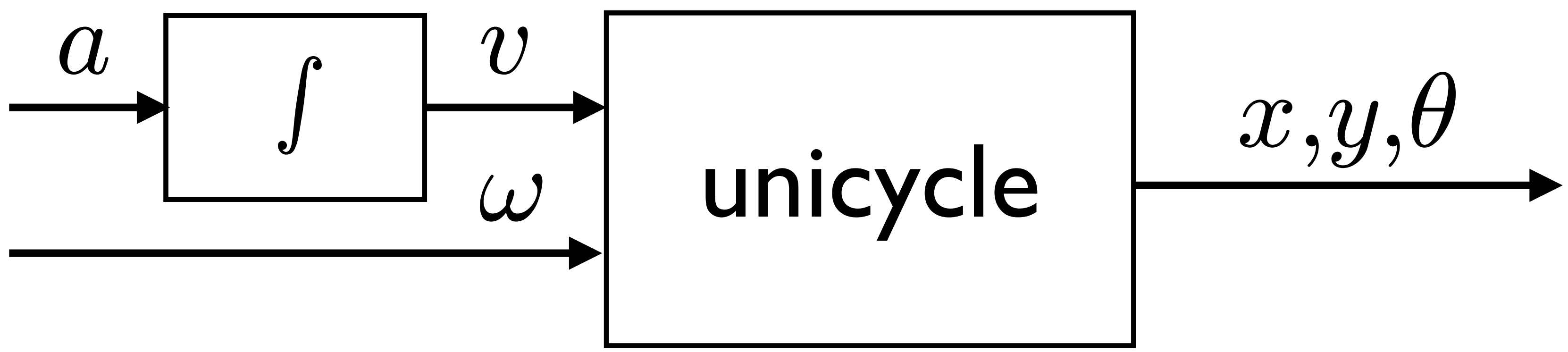


- a static controller based on output error
- needs $\dot{\boldsymbol{p}}_d$
- needs x,y,θ for output reconstruction and θ also for input transformation

via i/o linearization (dynamic feedback)

- rather than displacing the controlled output to B, we can keep the (flat) output (x,y) and achieve exact linearization by using a dynamic compensator
- to do this, perform a dynamic extension on the driving velocity channel (`slow down' the input)

$\dot{v} = a$ i.e.,



- we can now differentiate further the output

$$\ddot{x} = a\cos\theta - v\sin\theta\,\omega$$

$$\ddot{y} = a\sin\theta + v\cos\theta\,\omega$$

- the new decoupling matrix is nonsingular if $v \neq 0$

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \underbrace{\begin{pmatrix} \cos\theta & -v\sin\theta \\ \sin\theta & v\cos\theta \end{pmatrix}}_{\det = v} \begin{pmatrix} a \\ \omega \end{pmatrix} = \boldsymbol{T}(v,\theta) \begin{pmatrix} a \\ \omega \end{pmatrix}$$

- under this assumption, we may set

$$\begin{pmatrix} a \\ \omega \end{pmatrix} = \boldsymbol{T}^{-1}(v,\theta) \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\dfrac{\sin\theta}{v} & \dfrac{\cos\theta}{v} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

obtaining a fully linearized system

$$\ddot{x} = u_1$$
$$\ddot{y} = u_2$$

- achieve global exponential convergence of x, y to the desired trajectory by letting

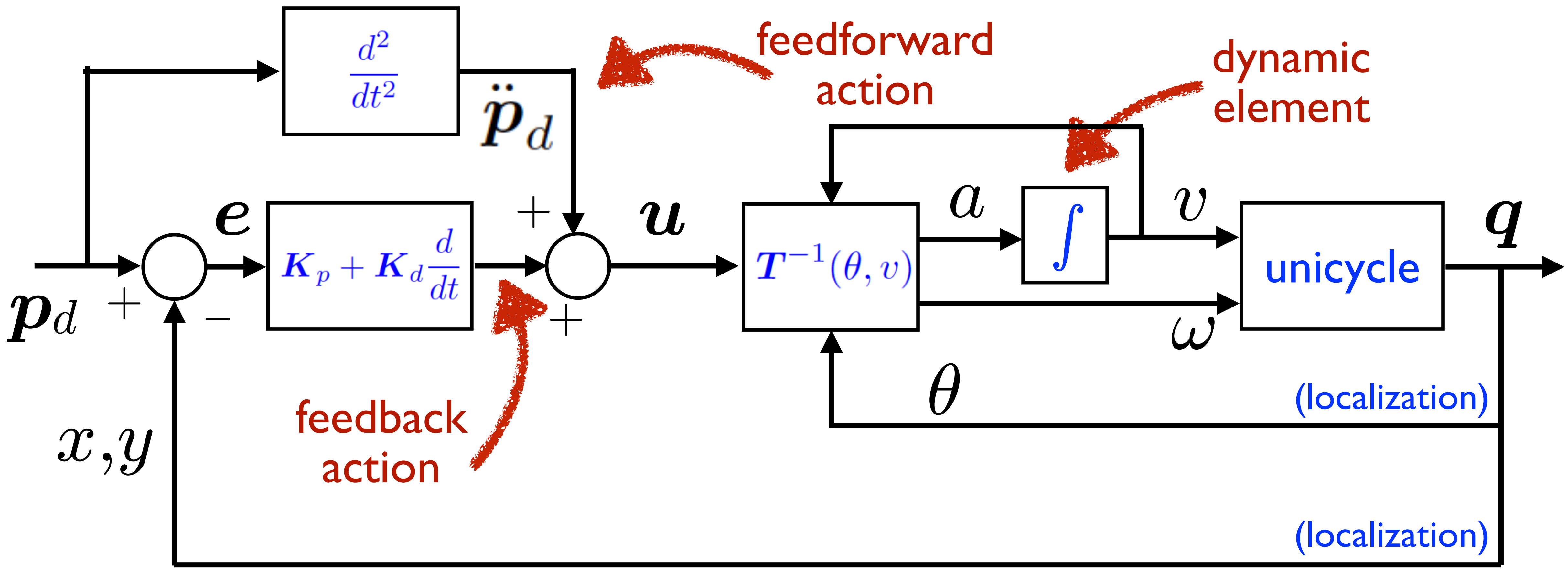
$$u_1 = \ddot{x}_d + k_{p1}(x_d - x) + k_{d1}(\dot{x}_d - \dot{x})$$
$$u_2 = \ddot{y}_d + k_{p2}(y_d - y) + k_{d2}(\dot{y}_d - \dot{y})$$

with $k_{p1}, k_{p2}, k_{d1}, k_{d2} > 0$

- since we are controlling the original flat output, there is no zero dynamics with this approach

- the desired trajectory must be twice differentiable and persistent, i.e., it must be $v_d \neq 0$ always (this singularity is structural in nonholonomic systems)

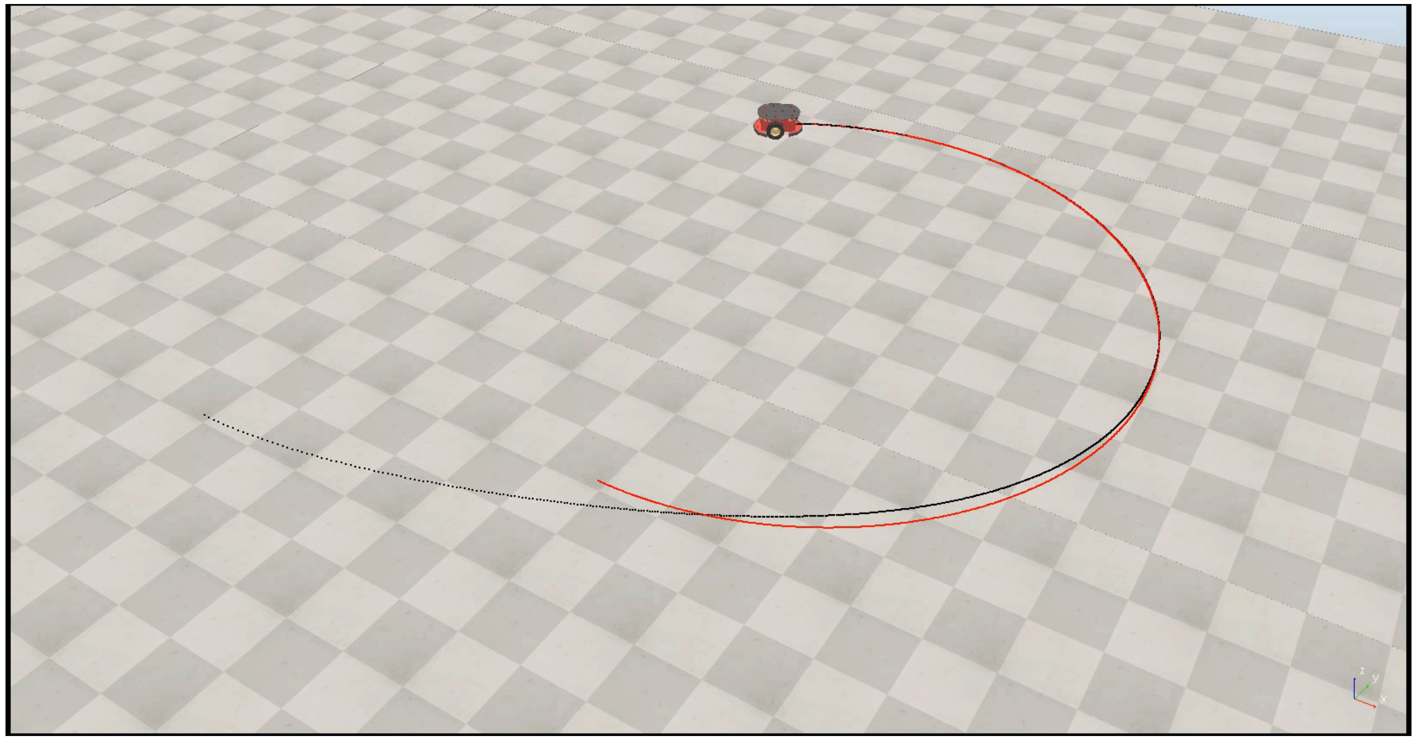
- the final block scheme for trajectory tracking via output error feedback + dynamic i/o linearization is

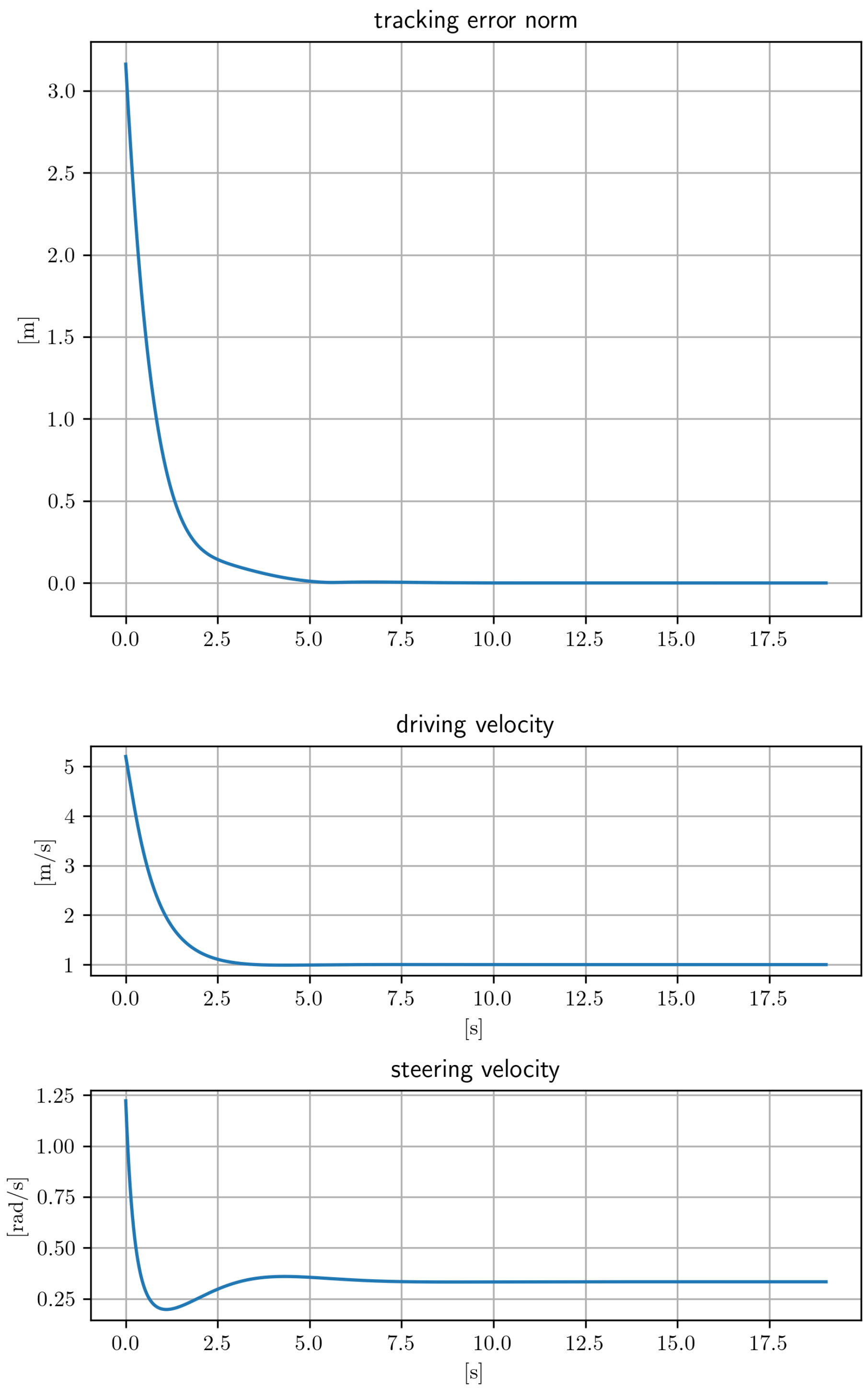


- a dynamic controller based on output error
- needs $\dot{\boldsymbol{p}}_d$ and $\ddot{\boldsymbol{p}}_d$
- needs x,y for error computation and θ for input transformation

simulations

tracking a circle via approximate linearization

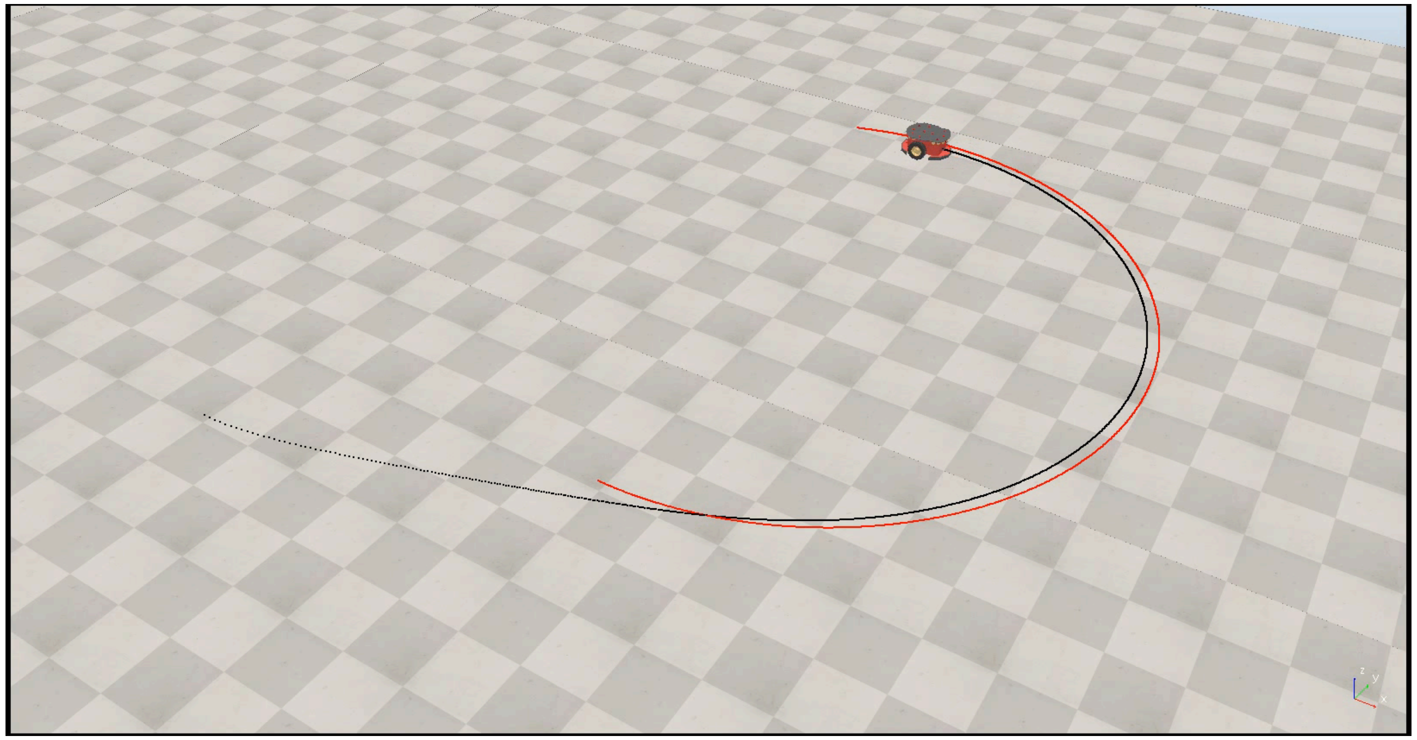


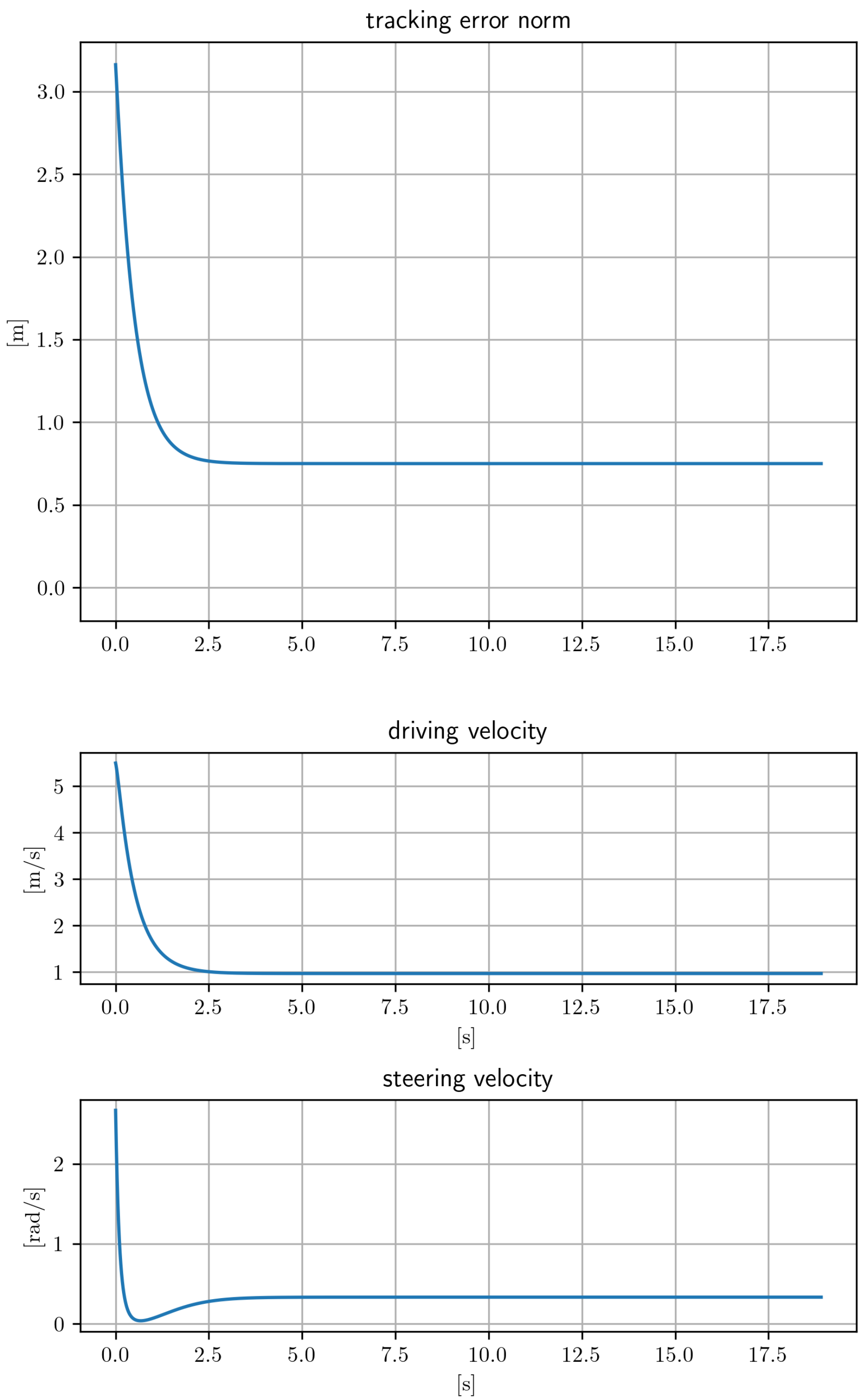


- only local stability is guaranteed

simulations

tracking a circle via static i/o linearization (b=0.75)

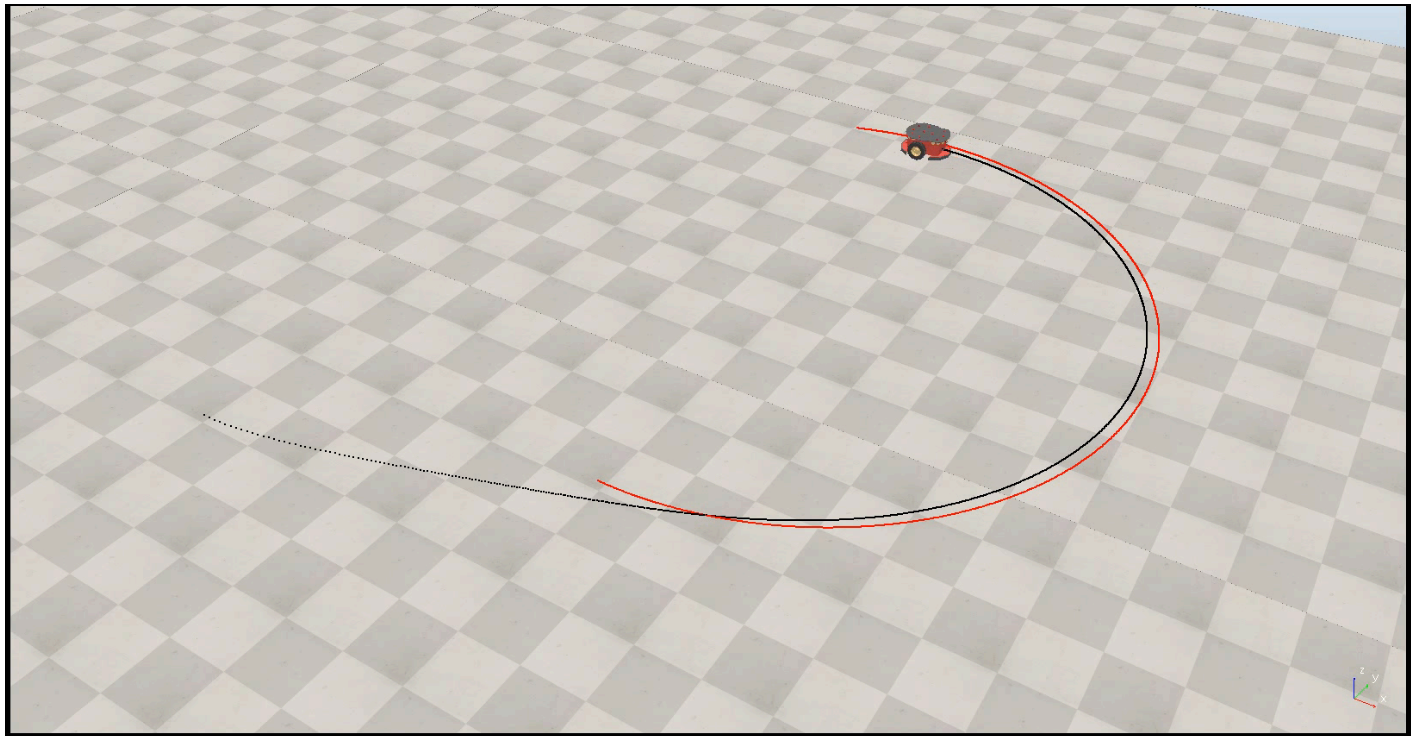


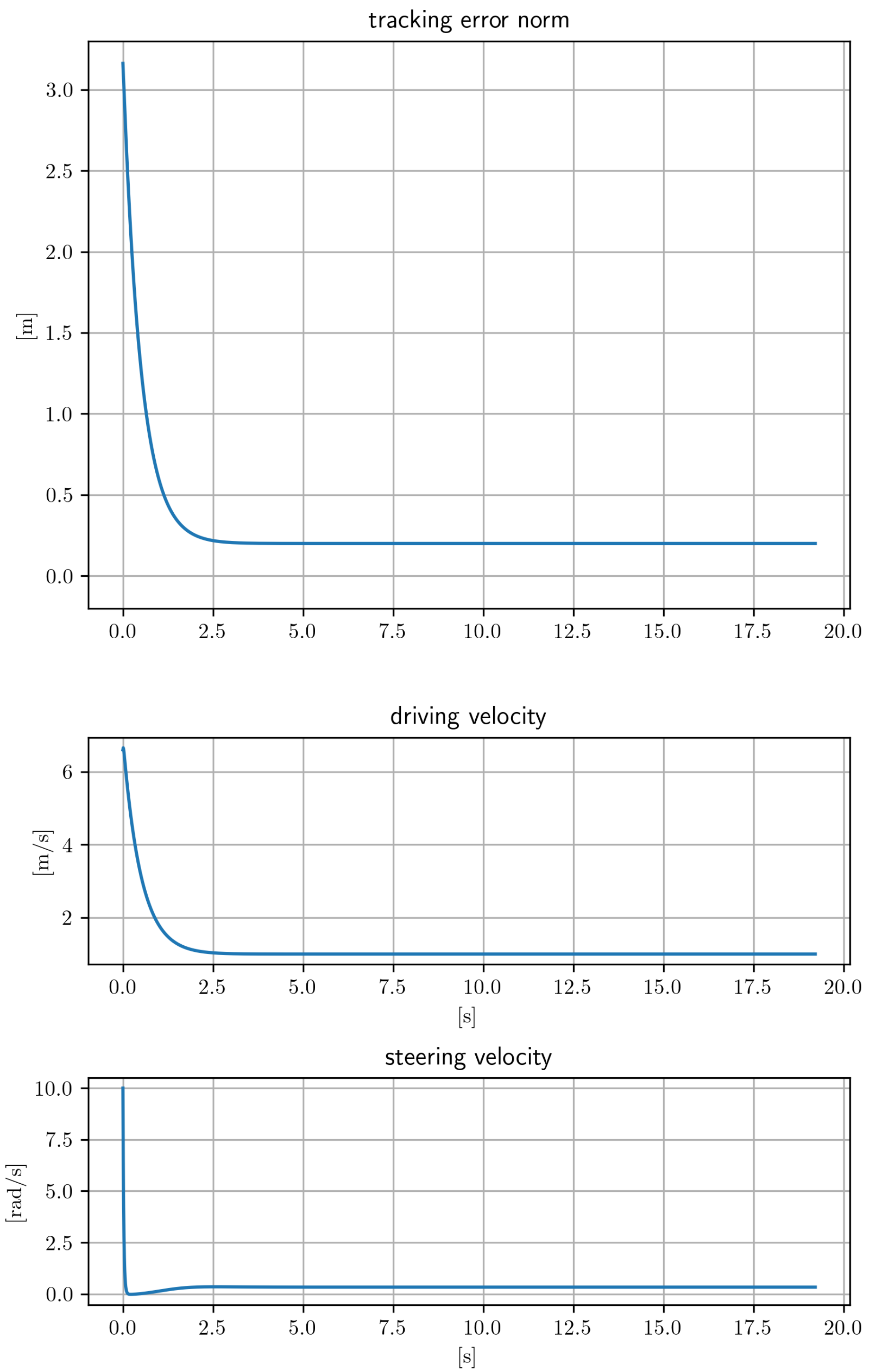


- steady-state error $= b$

simulations

tracking a circle via static i/o linearization ($b=0.2$)

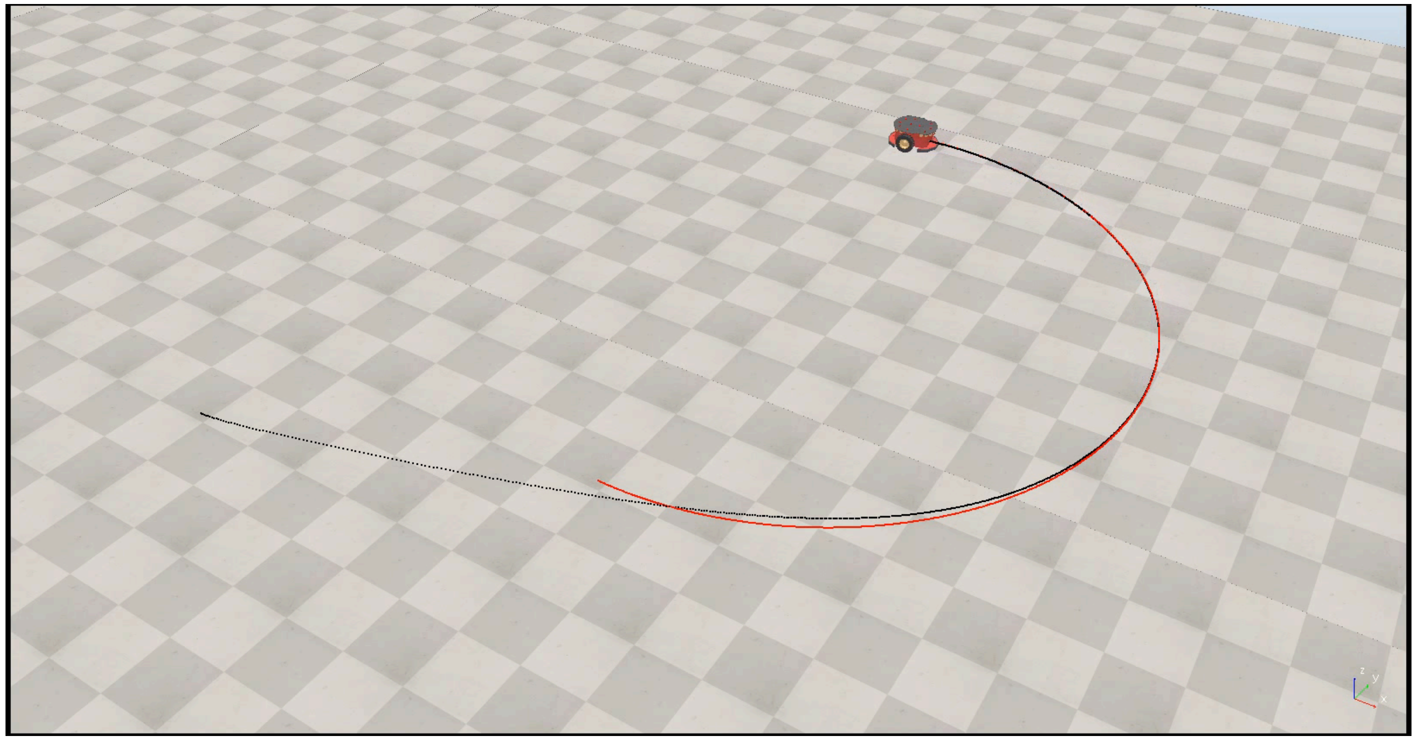


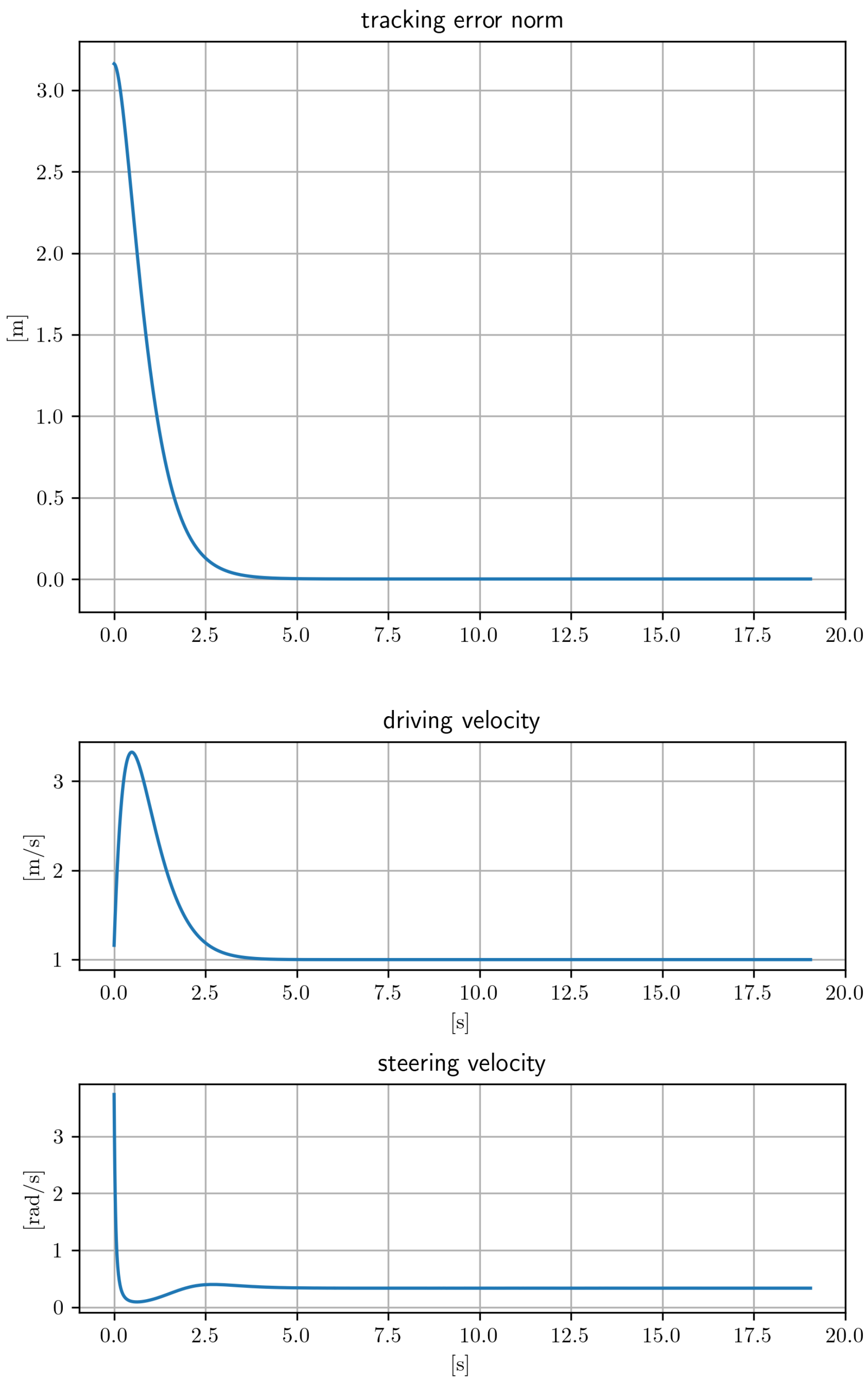


- steady-state error is now reduced but steering velocity increases

simulations

tracking a circle via dynamic i/o linearization

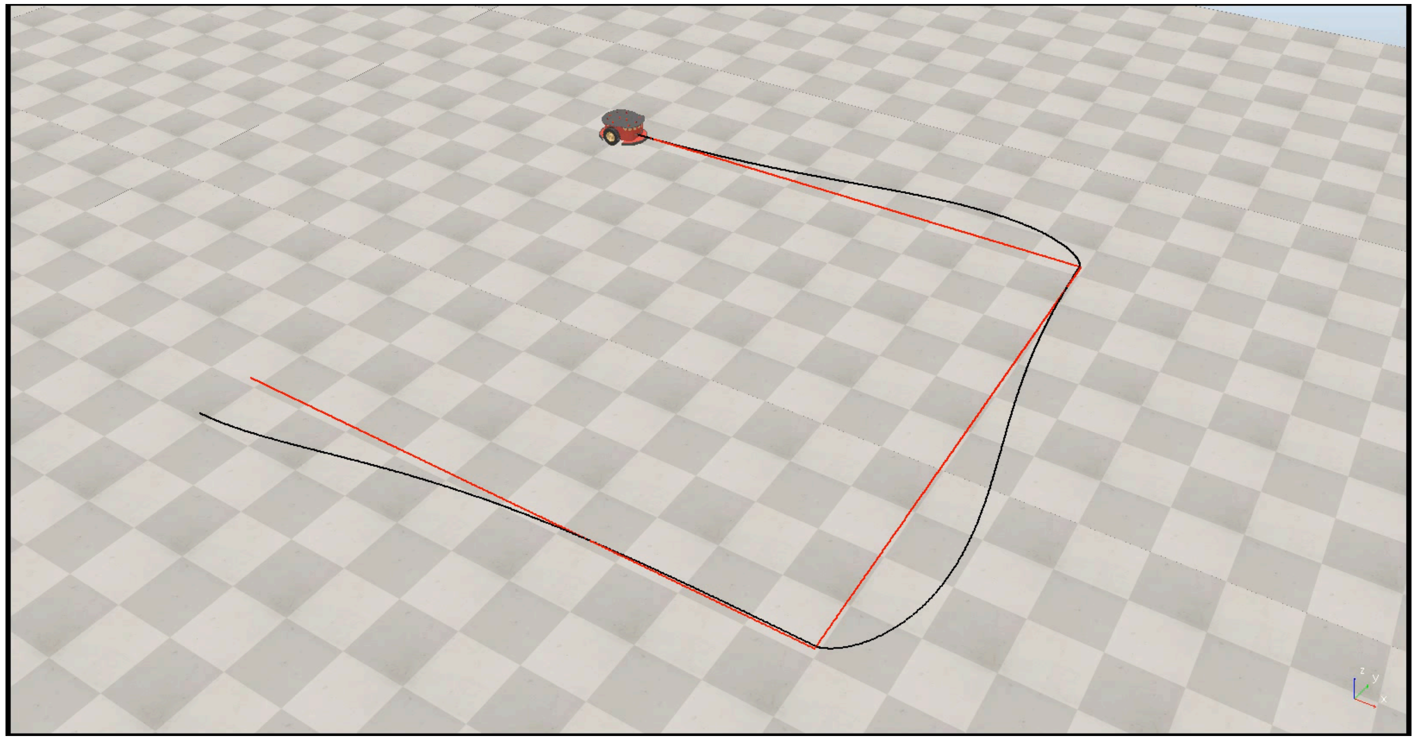


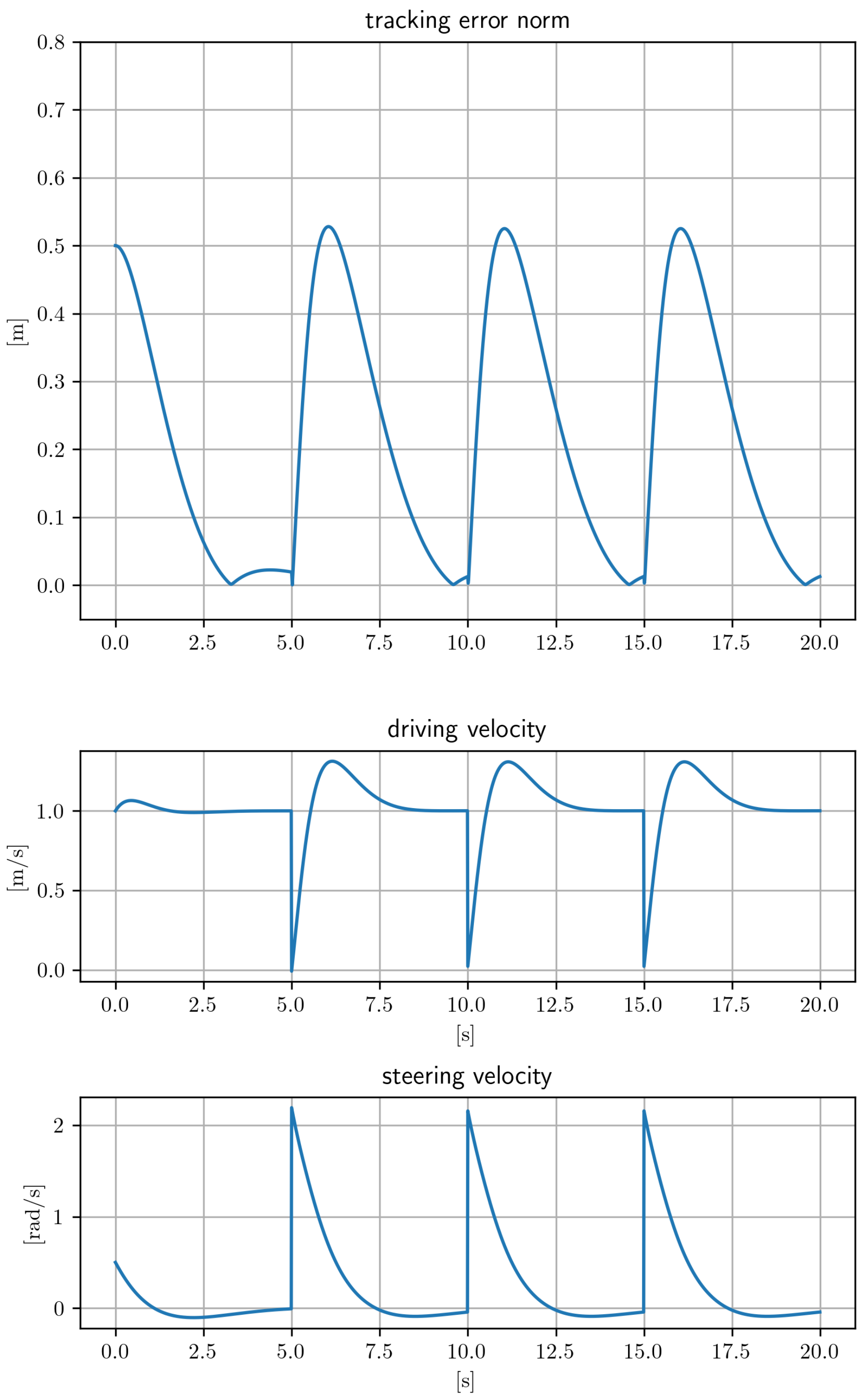


- zero steady-state error and reasonable velocities

simulations

tracking a square via approximate linearization

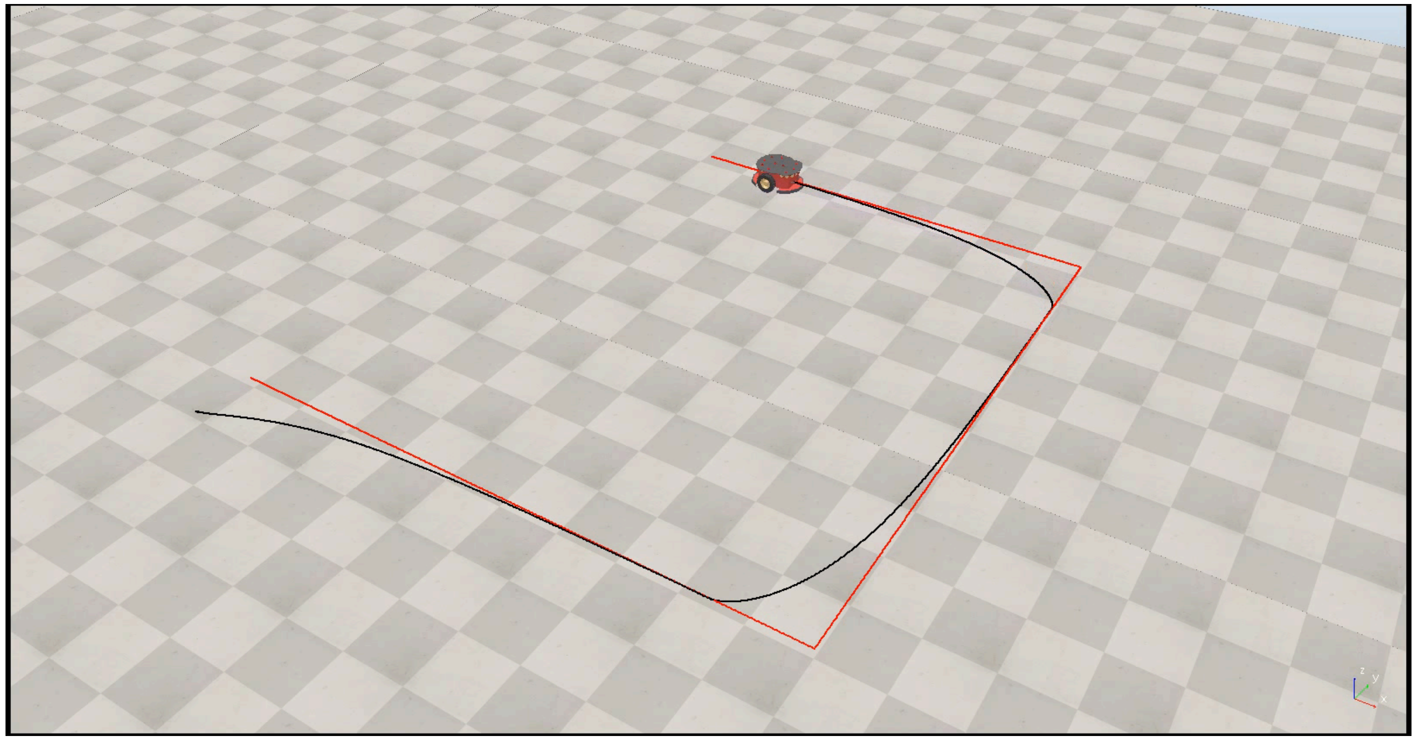


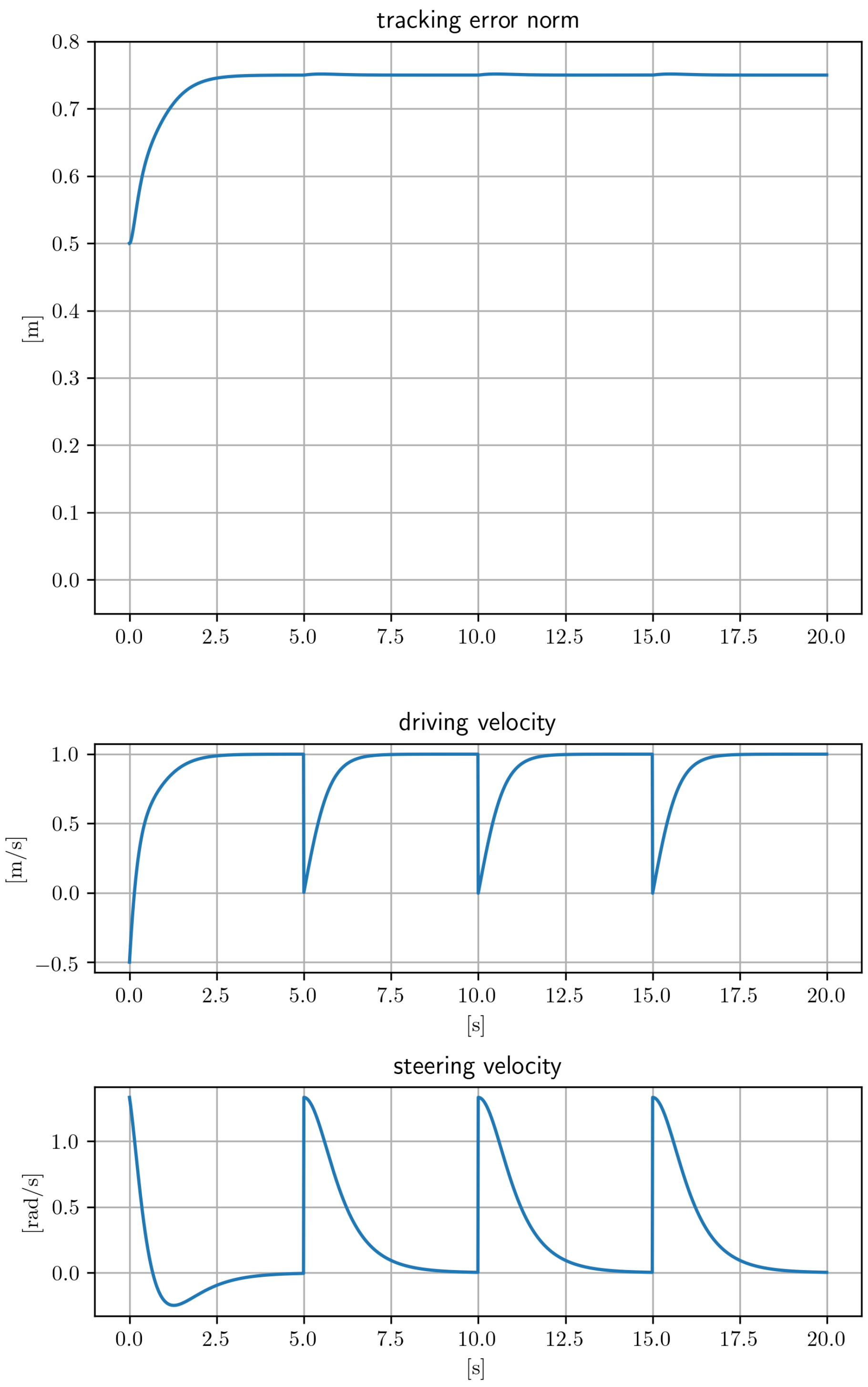


- only local stability is guaranteed
- a new transient at each corner

simulations

tracking a square via static i/o linearization (b=0.75)

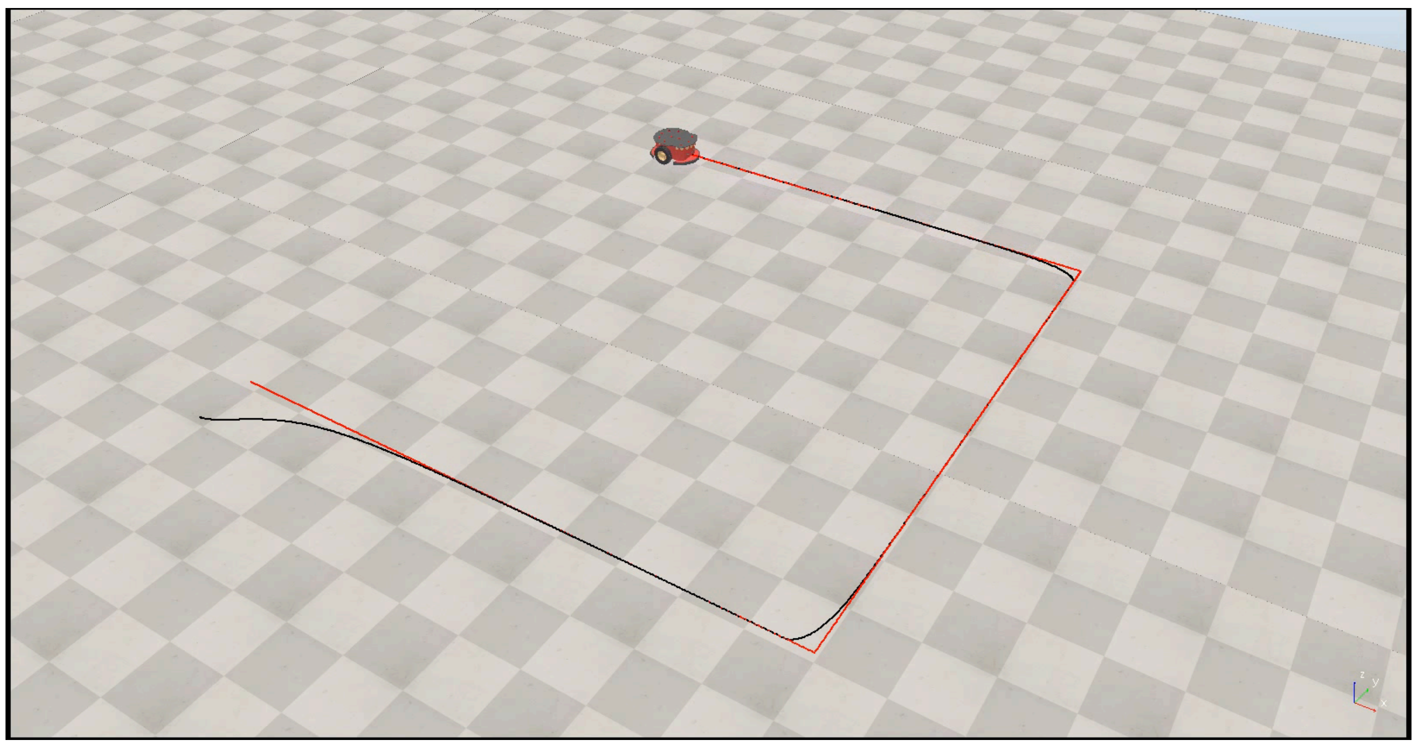


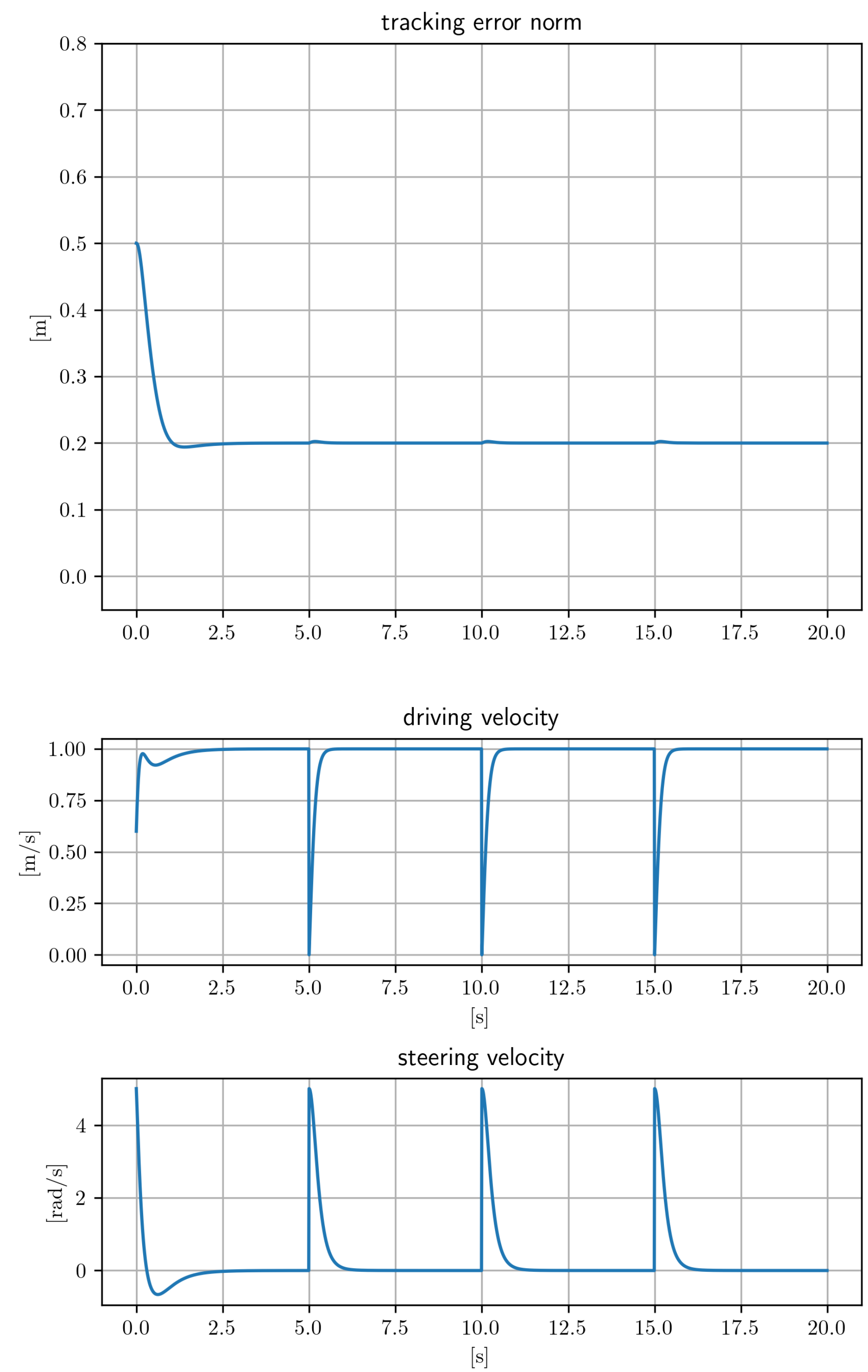


- steady-state error $= b$
- the displaced output provides a lookahead behavior

simulations

tracking a square via static i/o linearization (b=0.2)

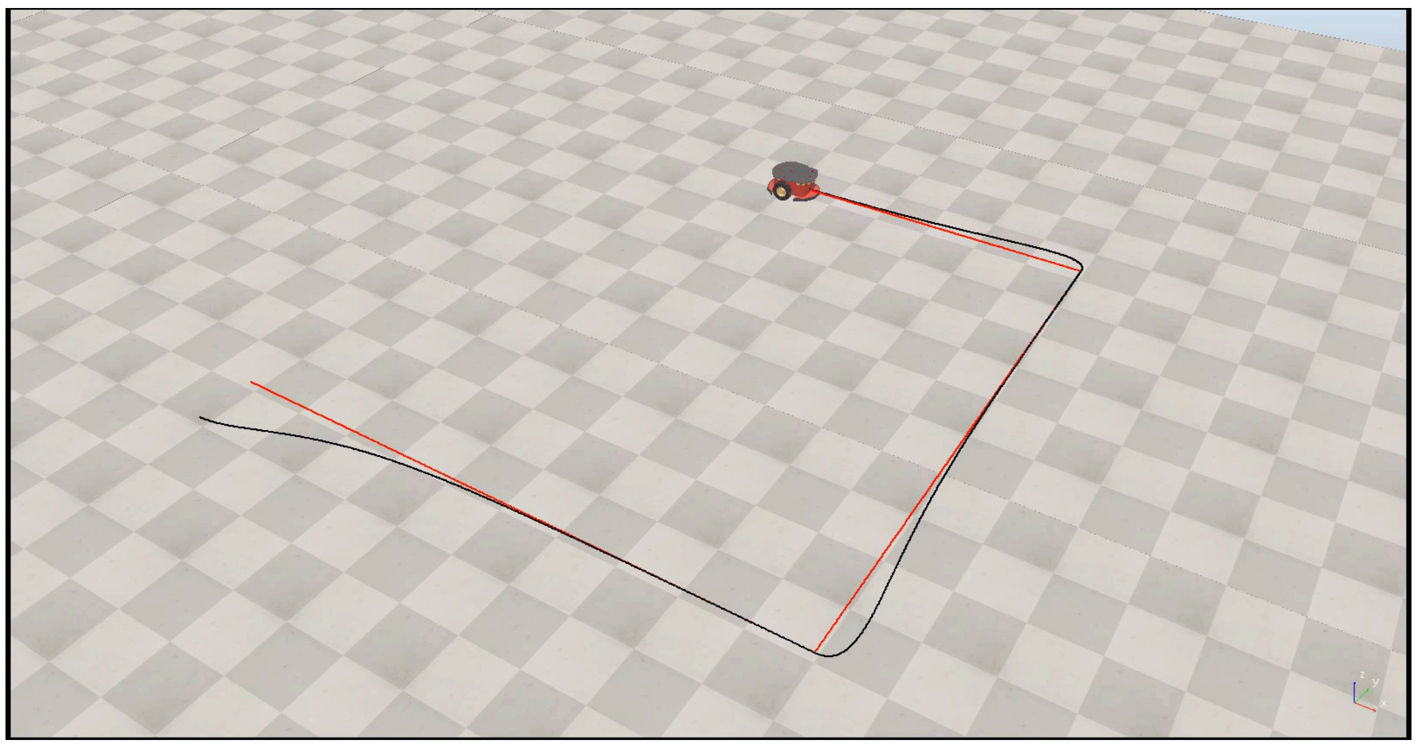


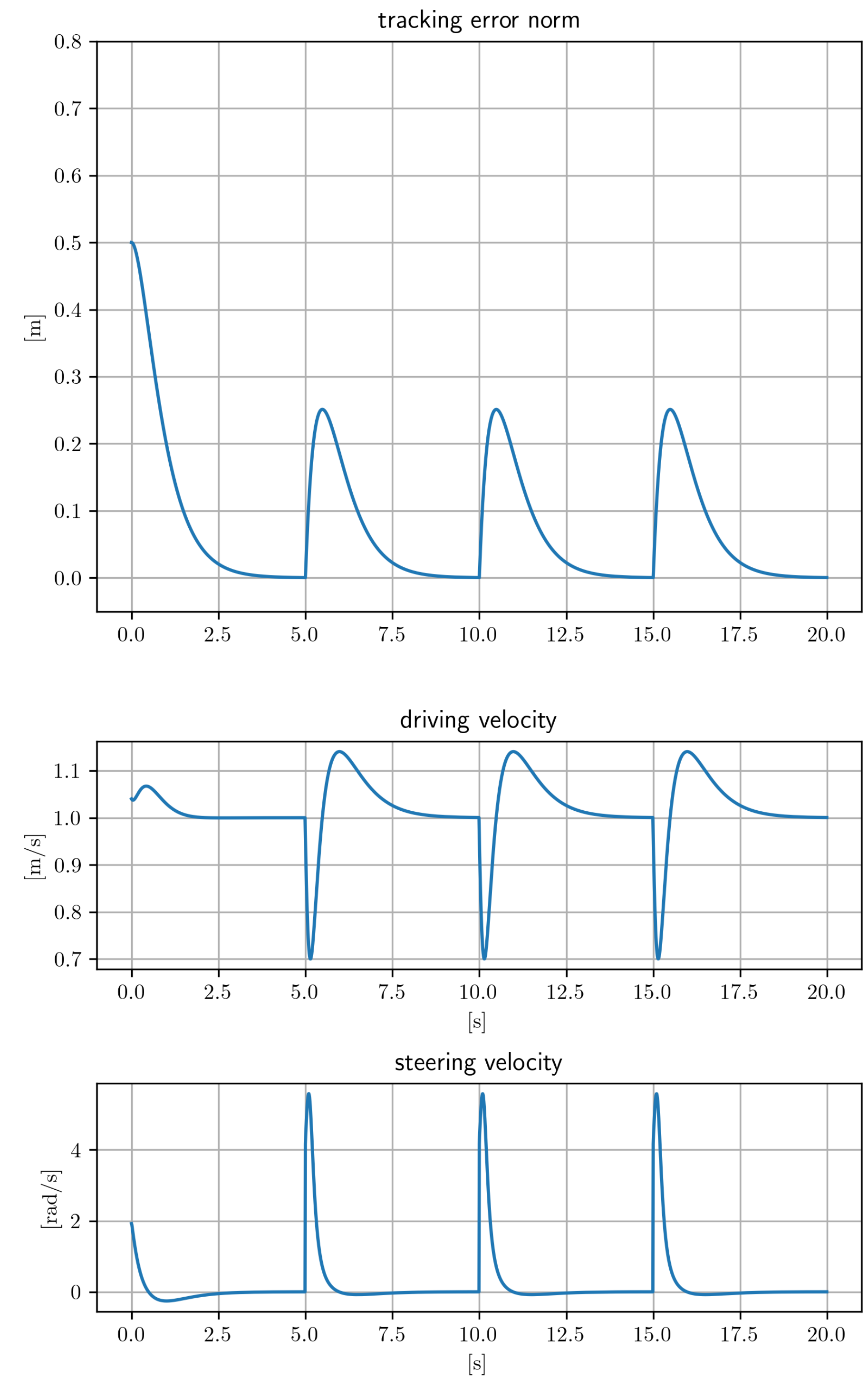


- steady-state error is now reduced but steering velocity increases

simulations

tracking a square via dynamic i/o linearization

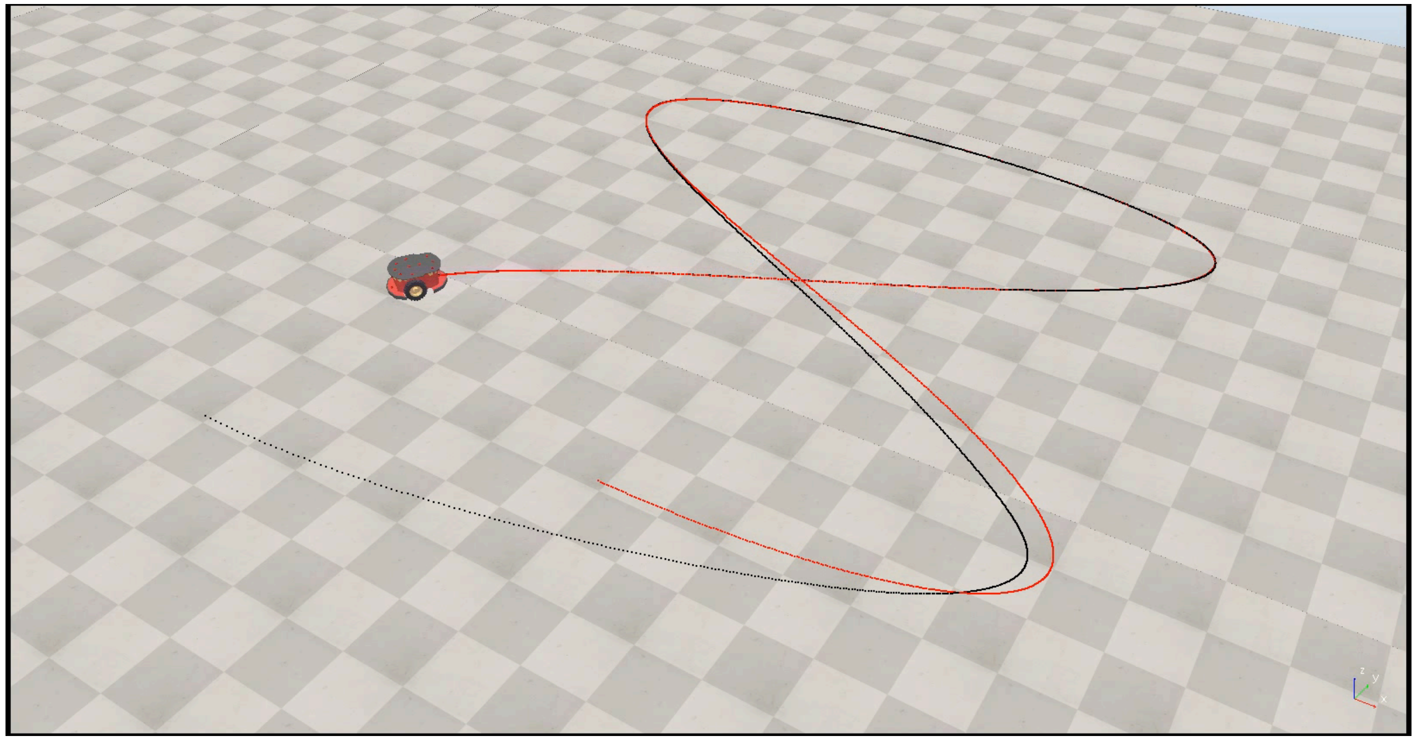


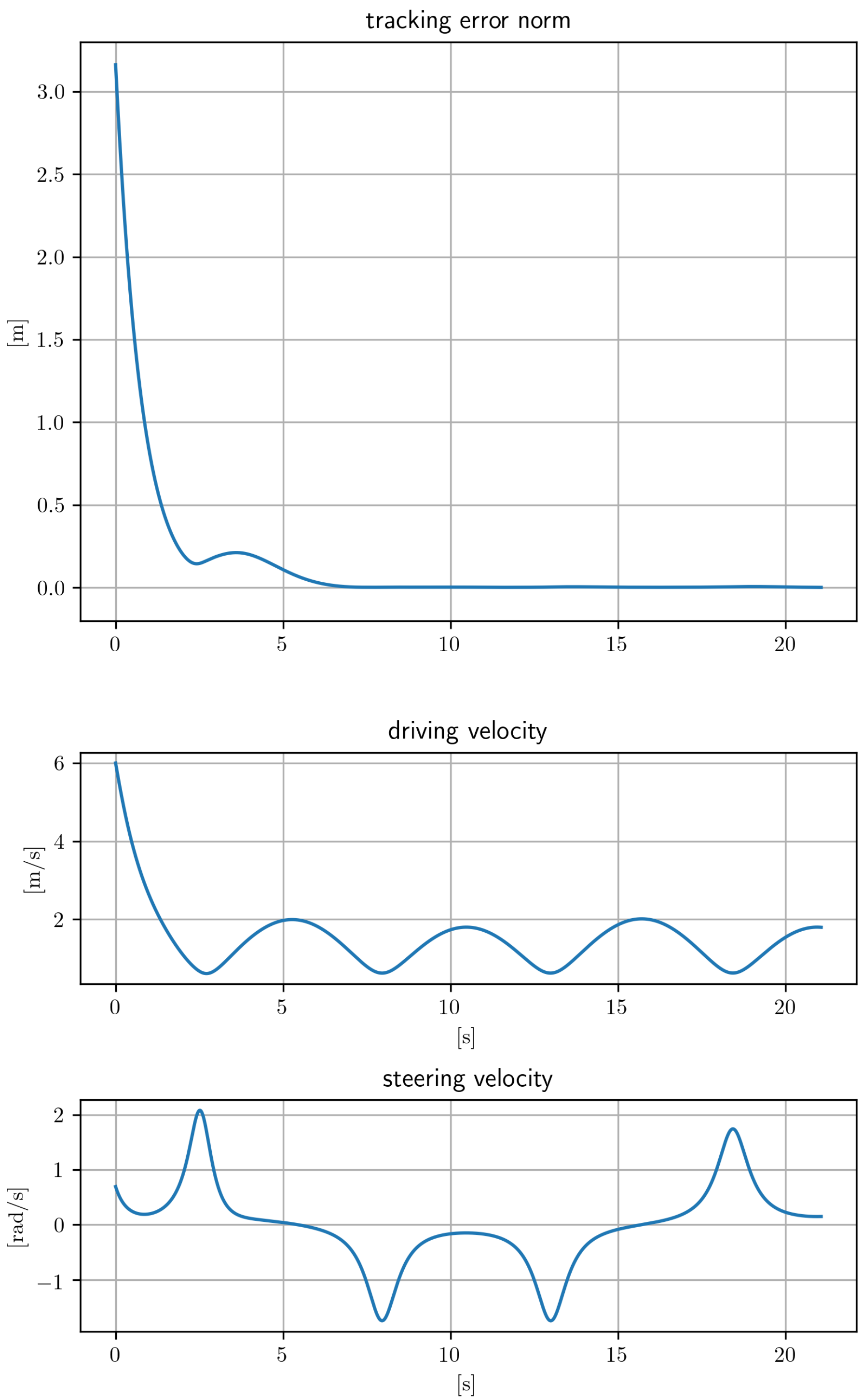


- faster transients and reasonable velocities

simulations

tracking a figure 8 via approximate linearization

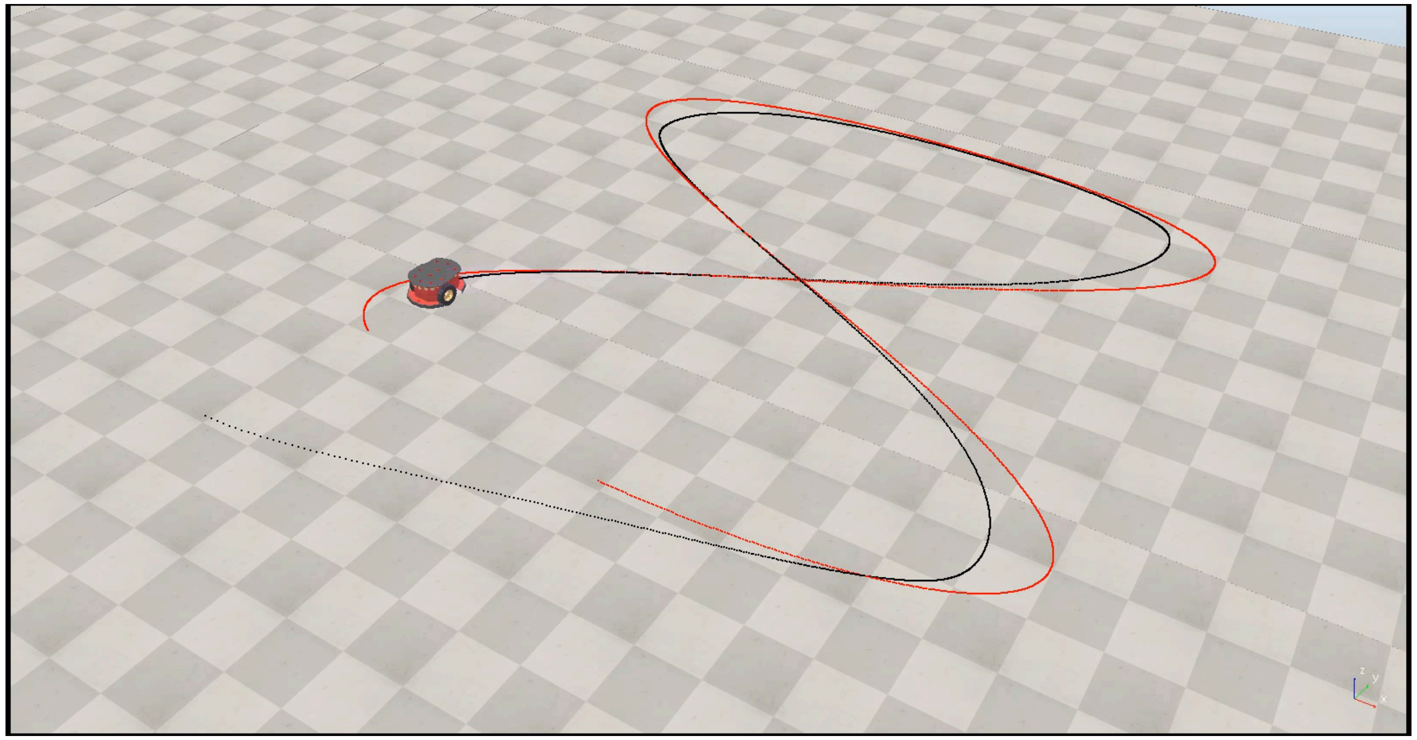


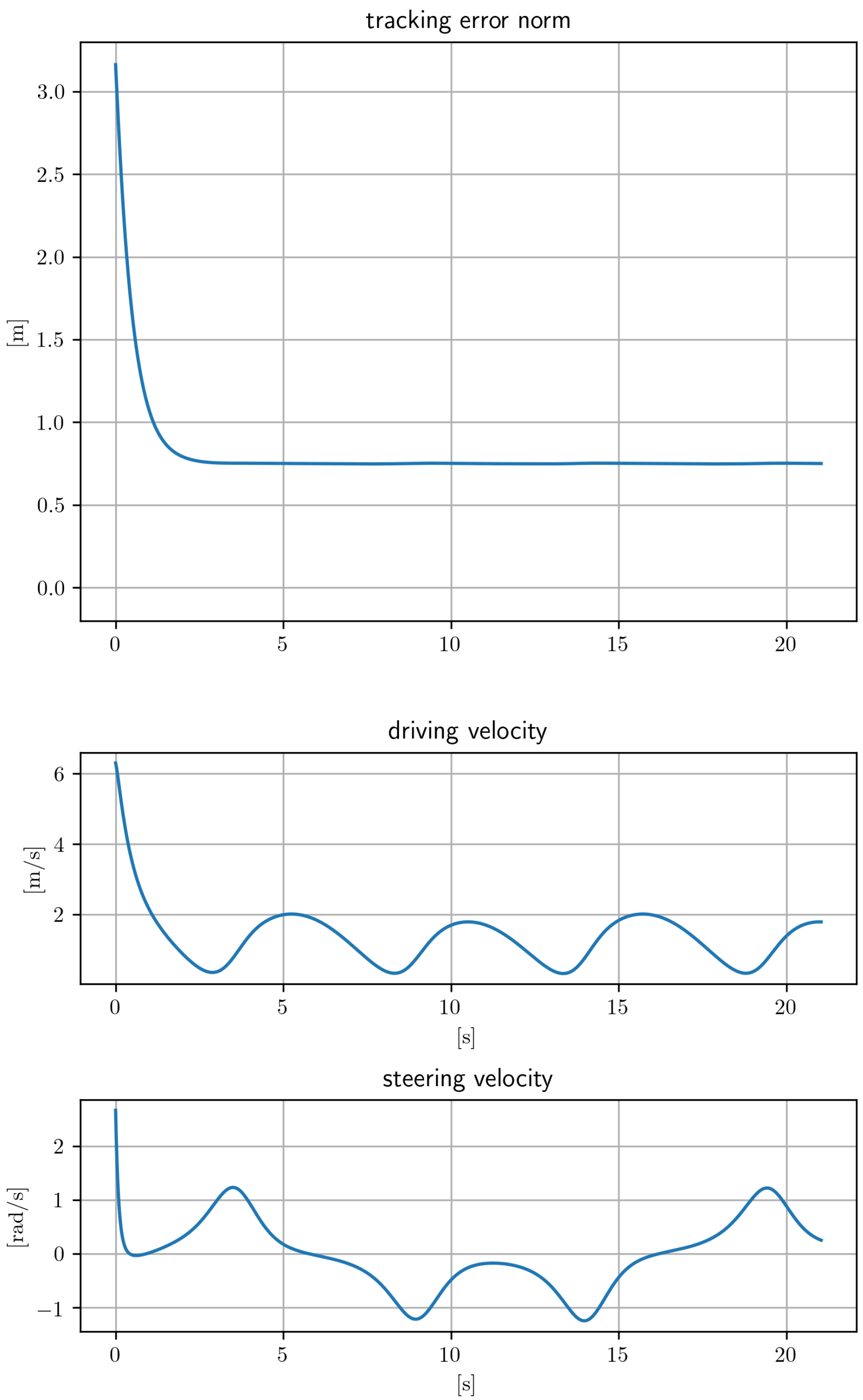


- local stability is not guaranteed, but performance is good

simulations

tracking a figure 8 via static i/o linearization (b=0.75)

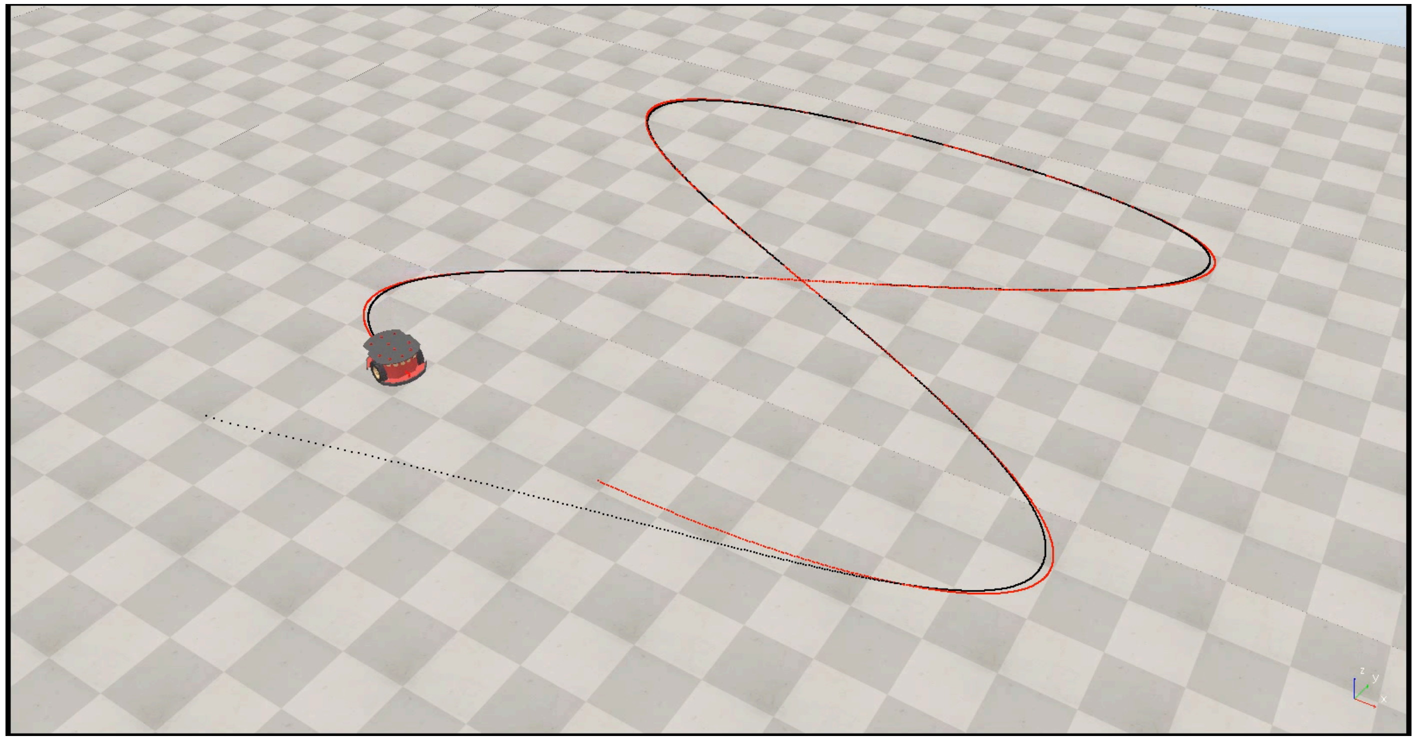


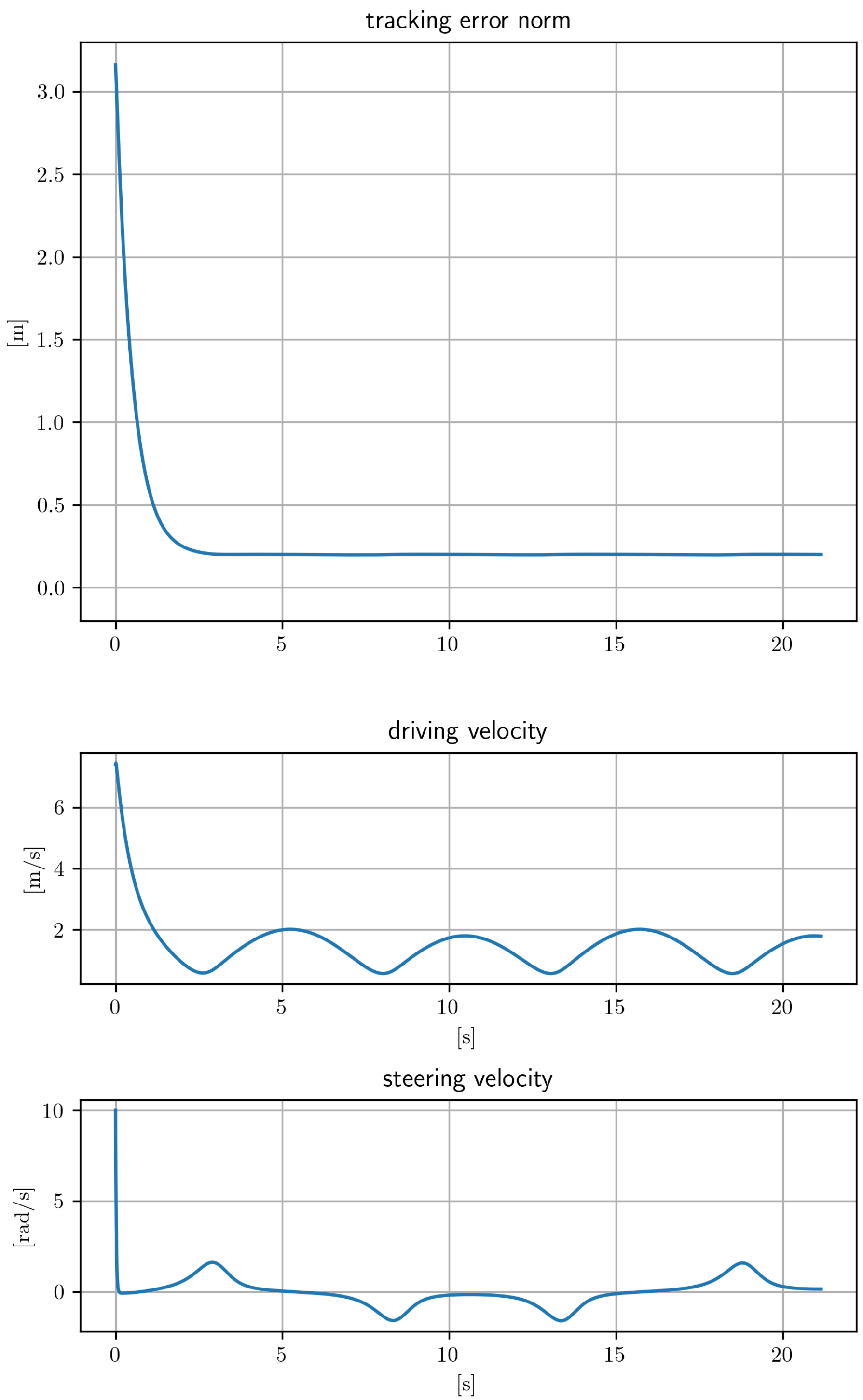


- steady-state error $= b$

simulations

tracking a figure 8 via static i/o linearization (b=0.2)

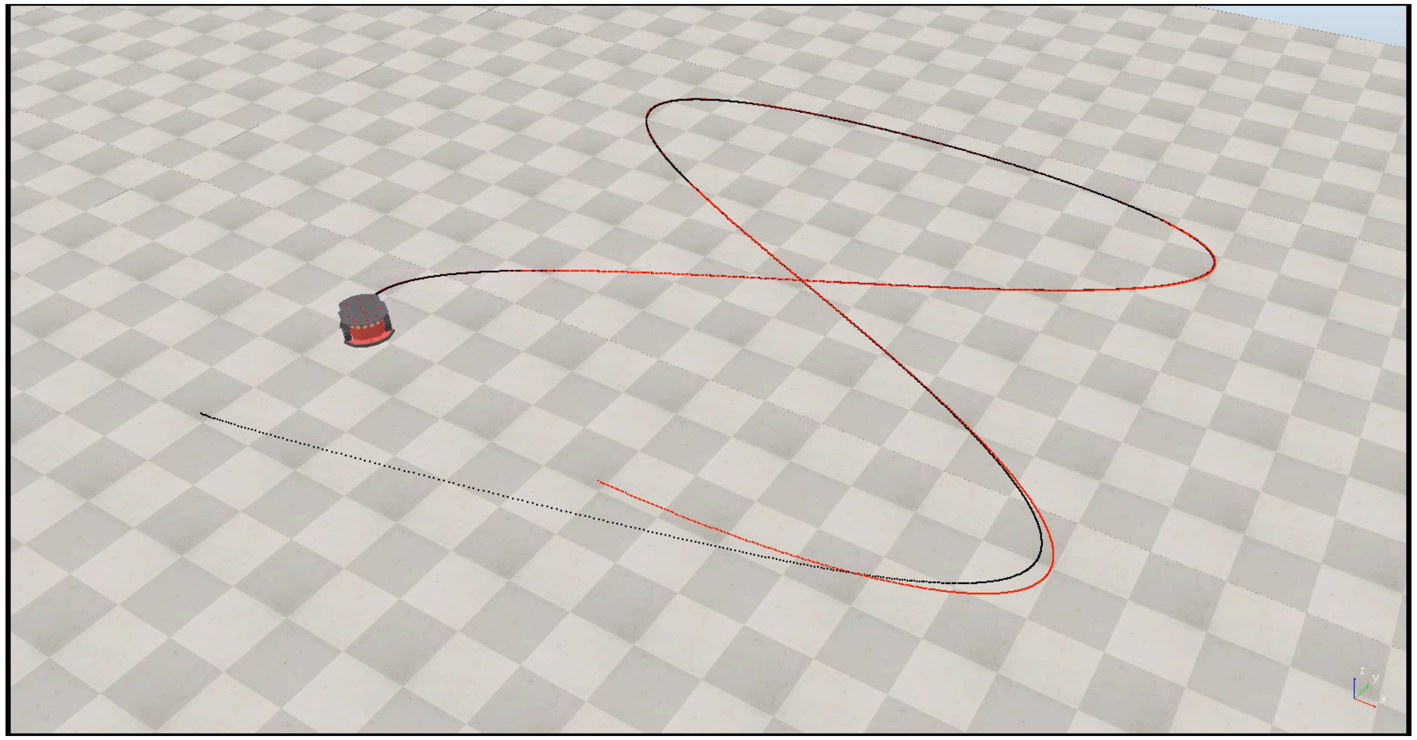


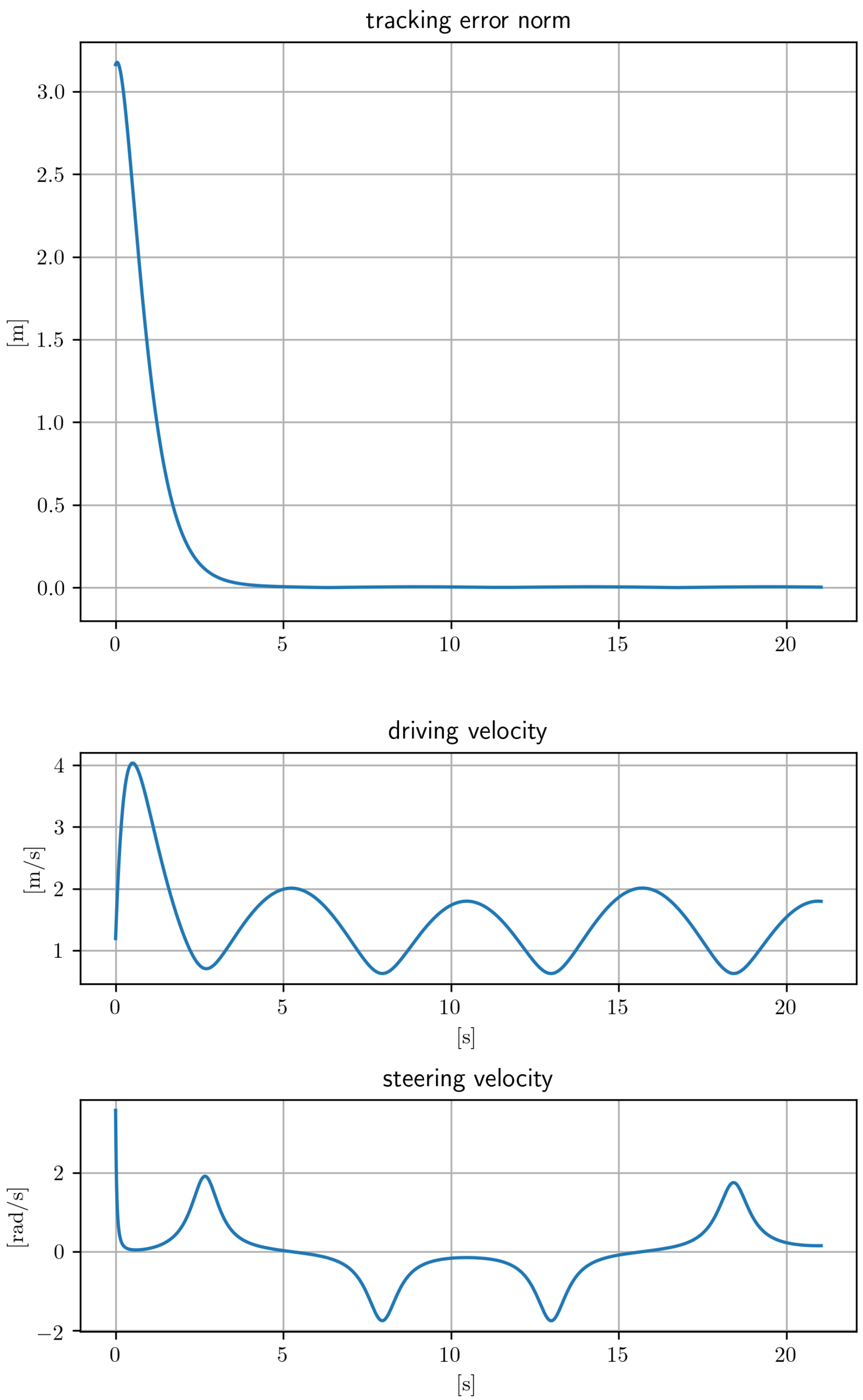


- steady-state error is now reduced but steering velocity increases

simulations

tracking a figure 8 via dynamic i/o linearization





- zero steady-state error and reasonable velocities